unexpected behavior of pyrcl in gui vs screen off mode Awaiting Agent

- S Sudhakar
- **Forum name:** #Test Automation

1. The pyrcl (py>3.6) script runs fine in gui mode. But the same script throws error when executed in screen off mode. Looks like "break" or "DO" commands don't work in pyrcl.
Ex: t32_instance.cmd('break')
Python trace:
File "/home/s0001673/.local/lib/python3.8/site-packages/lauterbach/trace32/rcl/_rc/_command.py", line 16, in __call__
self.__conn._cmd(cmd)
File "/home/msr/.local/lib/python3.8/site-packages/lauterbach/trace32/rcl/rcl.py", line 286, in _cmd
raise e.with_traceback(e.__traceback__) from None
File "/home/msr/.local/lib/python3.8/site-packages/lauterbach/trace32/rcl/rcl.py", line 284, in _cmd
self.__library.t32_executecommand(cmd.encode(), 4096)
File "/home/msr/.local/lib/python3.8/site-packages/lauterbach/trace32/rcl/_rc/_library.py", line 435, in t32_executecommand
raise CommandError(str(e), "command: ", cmd) from None
lauterbach.trace32.rcl._rc._command.CommandError: ('target system down', 'command: ', b'Break')
2. Also screen dumps using Printer function doesn't provide full/complete line. It cuts the info into half/quarter line of the actual output.

Comments (9)

Wiem Wala Benayed
1 year ago
1- It's normal that you can't execute 'Break' when your target is down, as evidenced by the message ''target system down''. The screen mode used has no impact on that. 2-In order to get the complete window contents, you have to manually configure it first: - Open your window - Adjust your window size (you should increase the columns) - Right click on the top left of your window - Choose "store command" - You will get a file with content similar to this : WinPOS 6.4286 3.4615 180. 26. 24. 1. W000 WinTABS 40. - Next time, before you do a WinPrint, you need to execute the first two line you get like that: WinPOS 6.4286 3.4615 180. 26. 24. 1. W000 WinTABS 40. WinPrint.

S **Sudhakar**
1 year ago
Bunch of thanks... For point 1: I can check in cmm that if system mode is running or not by invoking SYSTEM.MODE() == 0B (found from a demo script). But I'd like to know where are these functions documented with its return values for each mode? I couldn't find much help from "https://www2.lauterbach.com/pdf/ide_func.pdf". Are there better documents (or am I missing info?) that describe these functions? (sorry if something looks obvious. I'm just a beginner w.r.to T32 automation)
Wiem Wala Benayed
1 year ago
The function SYSTEM.MODE() is documented in this pdf: https://repo.lauterbach.com/pdf/general_func.pdf You can find all our pdf under /pdf
S **Sudhakar**
1 year ago
That's a good documentation. Also How can one read "the message line" for error handling purpose (given that some unexpected/expected errors arise? (also please point me to the right document as above) Ex: aborted SYStem.Mode Prepare: failed to establish communication or debug port fail.
Wiem Wala Benayed
1 year ago
You can work with try and except blocks, so in the exception, you can execute the following API functions: - CommandError : to check if a command returns an error, for example : try: self.cmd("DO {}".format(cmd)) except CommandError as e: raise PracticeError(str(e)) from None - FunctionError : to check if a function returns an error, for example : try: result_value, result_type = self.__library.t32_executefunction(func) except FunctionError as e: raise e.with_traceback(e.__traceback__) from None Most of the errors are documented here: https://repo.lauterbach.com/pdf/error.pdf
S **Sudhakar**
1 year ago
How to inject/simulate any errors in T32 to make the code robust for any run time errors? some errors are easy to produce like "file not found". Any help here?
Wiem Wala Benayed
1 year ago
You can use the PRACTICE command ECHO with %error format: ECHO %error "file not found"
S **Sudhakar**
1 year ago
How to implement ON ERROR mechanism in python (without using cmm)? In other words, How to define what to do when error occurs in python script? I couldn't find much details in pyrcl documentation? Any insights on this?
Wiem Wala Benayed
1 year ago
You can use this example: from typing import Any import lauterbach.trace32.rcl as t32 def print_notification(*args: Any) -> None: print("notification:", args) dbg = t32.connect() dbg.library.t32_notifyeventenable("SYSUP", print_notification) dbg.library.t32_checkstatenotify(1) In RCL version 1.0.9, you will get an error. There is a small fix that need to be done in function t32_checkstatenotify of \lauterbach\trace32\rcl_rc_library.py: if event_type == T32_E_ONEVENT: offset=16 event_name = msg_data[offset:].decode().rstrip("\0")