Not able to read structure variable (pyrcl) Open Discussion

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

Hi, I'm trying to read a struct using pyrcl.
Py line: var_read = connection.variable.read("var_name")

Traceback:
File "/home/s0001673/.local/lib/python3.10/site-packages/lauterbach/trace32/rc
l/_rc/_variable.py", File "/home/s000
return self.read_by_name(name, **kwargs)
File "/home/s0001673/.local/lib/python3.10/site-packages/lauterbach/trace32/rc
l/_rc/_variable.py", File "/home/s0001673/.lo
raise VariableError(result.err_code, result.err_msg)
lauterbach.trace32.rcl._rc._variable.VariableError: (65535, '')

When I observed the library, it's due to payload becoming empty/none (see the attached image).

But with the following py line, I'm able to read the var.
var_read = connection.fnc("Var.VALUE(var_name)")

when I tried to understand what's "t32_exp()" (as it causes error), I couldn't find any resource. Could you help here?

Comments (2)

S **Sudhakar**
2 years ago
I'm sorry. It is an array instance (not struct). But I am able to read with indexing (eg: var_name[2]), not as a whole. I think it's same with struct as well if I understand correctly

Anonymous
2 years ago

Hello,

Reading all elements of a variable without specifying an index isn't feasible.

If you attempt to omit the index to read all elements at once, you will encounter an error, as you have already experienced.

As a workaround, if you wish to obtain the entire content of a variable, you may need to iterate over its elements programmatically,

as demonstrated in the example below:

```
import lauterbach.trace32.rcl as t32

dbg = t32.autoconnect()

bit_count = 19

bits_string = ""

for i in range(bit_count):

current_bit = dbg.variable.read("flags[" + str(i) + "]").value

bits_string += str(current_bit).zfill(2)

print(bits_string)
```

The above example will allow you to retrieve all the elements of a certain variable.

Another method you can use to retrieve values of a variable, regardless of the data type (such as structs),

is by sending the TRACE32 command 'var <Your Variable>' and then retrieving the displayed value from the Area window, as shown in the example below:

```
import lauterbach.trace32.rcl as t32

dbg = t32.autoconnect()

dbg.cmd('var vbfield')

message = dbg.fnc('AREA.LINE("A000",0)')

print(message)
```

Best Regards,

Firas ZOUAGHI