Real-time DataLogger configuration

PLCnext Technology
Designed by PHOENIX CONTACT

START

Session(s)

Log variables from a single or several tasks?

- several tasks →
- single task ↓

All values / events in the same file is useful for comparing values to the same time stamp. Different files provide more clarity when logging many variables.

Log values / events in the same file or in different files?

- same file
- different files →

From firmware 2021.3 on, the task can be addressed by its name with the taskContext attribute (instead of samplingInterval)

Create a session (= config file) reflecting that task, e.g.:
task cycle = 10ms:
samplingInterval <= task cycle time (10ms)

Create a single session (= config file) combining logged data of several tasks.

Example:
task1: task_cycle_1=5ms
task2: task_cycle_2=10ms
task3: task_cycle_3=20ms

Create separate sessions (= config files) for each task.

- use case 1
- use case 2

For logging values with all changes, the respective task cycle time should be a multiple of the sampling interval.
E.g. for logging task1 with all changes:
samplingInterval <= min. task cycle time (n x 5ms)

For logging values task-synchronous to one of the task cycles, its task cycle time should be higher than the sampling interval.
E.g. for logging synchronous to task2:
task_cycle_2 (10ms) >= samplingInterval > task_cycle_1 (5ms)

- task_cycle_1 = session 1
- task_cycle_2 = session 2
- task_cycle_3 = session 3

`<General name="test-session" samplingInterval="10ms"/>`
or from firmware 2021.3 on
`<General name="test-session" taskContext="mytask"/>`

`<General name="test-session" samplingInterval="5ms"/>`

`<General name="test-session" samplingInterval="10ms"/>`

`<General name="test-session1" samplingInterval="5ms"/>`

`<General name="test-session2" samplingInterval="10ms"/>`

`<General name="test-session3" samplingInterval="20ms"/>`

If there are **low CPU resources but sufficient RAM capacity** then write less frequently into the data sink:
publishInterval=" T x 1000ms" bufferCapacity="2000"

After loading the application, how are the CPU and RAM resources?

If there are **sufficient CPU resources** then write more frequently into the data sink:
publishInterval="T x ms" bufferCapacity="2"

If both is sufficient try a balanced RAM –CPU proposition

`<General name="test-session(x)" samplingInterval ="T x ms"` **publishInterval="250ms" bufferCapacity="2000"/>**

Datasink(s)

If the controller is switched off or is restarted **the data will be lost.** Variables of this sink type can only be queried using Remote Service Calls (RSC). Access via file system is not possible.

`<Datasink type="volatile"/>`

Log variables volatile in the RAM or by writing into a file?

- RAM
- file →

`<Datasink type="db" dst="/opt/plcnext/DataSink.db"/>`

Log values / events in the same file or in different files?

- same file
- different files →

Keep the **deleteRatio** percentage lower with larger **maxFileSizes**!

`<Datasink type="volatile"` **rollover="false" maxFileSize="5000000" deleteRatio="5"/>**

`<Datasink type="db" dst="/opt/plcnext/DataSink.db"` **rollover="false" maxFileSize="5000000" deleteRatio="5"/>**

`<Datasink type="db" dst="/opt/plcnext/DataSink.db"` **rollover="true" maxFiles="5" maxFileSize="5000000"/>**

Log only changed values or all values?

- only changed values
- all values

`<Datasink type="volatile" rollover="false" maxFileSize="5000000" deleteRatio="5" storeChangesOnly="true"/>`

`<Datasink type="volatile" rollover="false" maxFileSize="5000000" deleteRatio="5" storeChangesOnly="false"/>`

`<Datasink type="db" dst="/opt/plcnext/DataSink.db" rollover="false" maxFileSize="5000000" deleteRatio="5" storeChangesOnly="true"/>`

`<Datasink type="db" dst="/opt/plcnext/DataSink.db" rollover="false" maxFileSize="5000000" deleteRatio="5" storeChangesOnly="false"/>`

`<Datasink type="db" dst="/opt/plcnext/DataSink.db" rollover="true" maxFiles="5" maxFileSize="5000000" storeChangesOnly="true"/>`

`<Datasink type="db" dst="/opt/plcnext/DataSink.db" rollover="true" maxFiles="5" maxFileSize="5000000" storeChangesOnly="false"/>`

Variable(s)

```
<Variables>
  <Variable name="var_1"/>
  <Variable name="var_2"/>
  ...
  <Variable name="var_n"/>
</Variables>
```

max. 996 variables per session

END