

Load Script

Start Script

Debug Script

Edit Script

Sequencer File

Filename: **annealing22/JouleMeth/monitor_Joule.msl** Comment: **HV scan run with LED (59.0V, 54.0 to 57.0V with every 0.2V)**

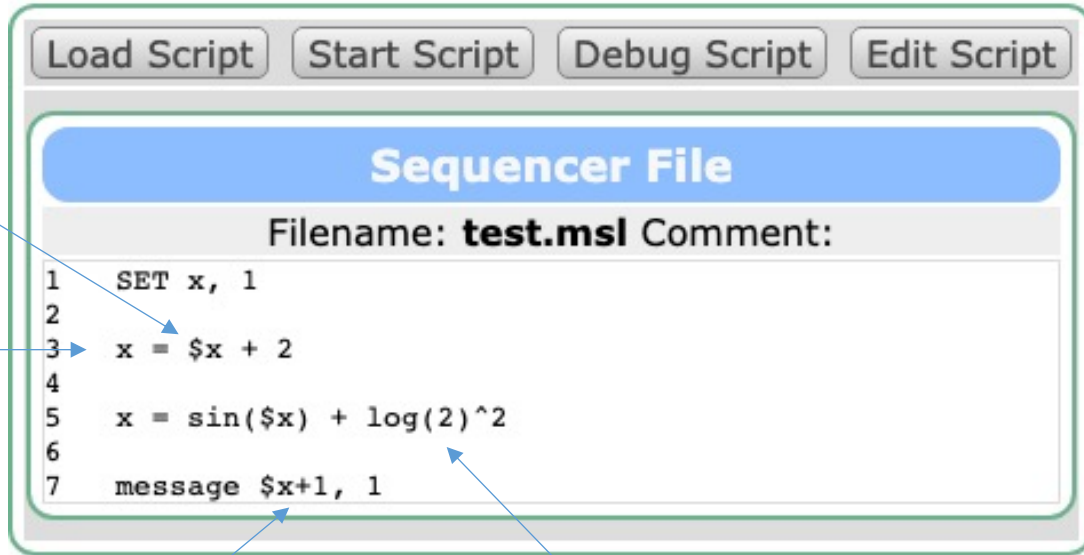
```
1 COMMENT "HV scan run with LED (59.0V, 54.0 to 57.0V with every 0.2V)
2 #INCLUDE /home/meg/online/sequencer/lib/WDAQPMT
3 INCLUDE /home/meg/online/sequencer/lib/WDAQMPPC
4 #INCLUDE /home/meg/online/sequencer/lib/WDAQTC
5 #INCLUDE /home/meg/online/sequencer/lib/WDAQCDCH
6 INCLUDE /home/meg/online/sequencer/lib/WDAQSystem
7 INCLUDE /home/meg/online/sequencer/annealing22/ledcontrol
8 INCLUDE /home/meg/online/sequencer/annealing22/setswzerosuppression
9
10 # Run parameters
11 SET Nevent, 1000
12 SET Ampl, 0.90
13
14 SET PMTHVConfig, 210815
15
16 # ADC/DRS_ON/OFF.
```

MIDAS Sequencer Update

Stefan Ritt

19.04.2022

Math added



Access variables by "\$"

SET can be replaced by simple "="

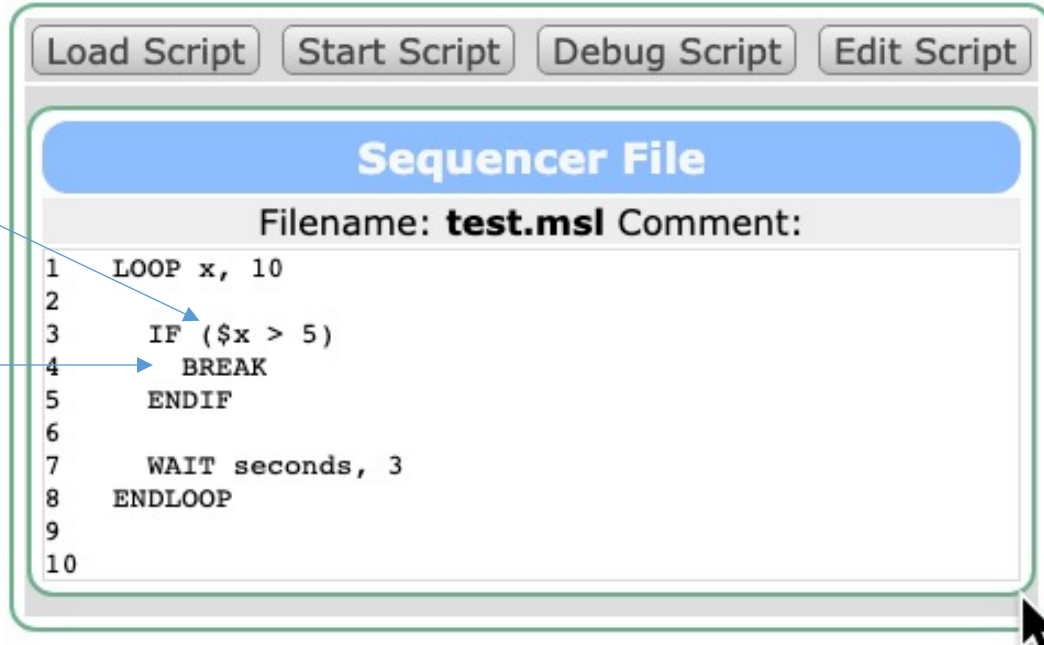
sin(), abs(), floor(), log(), pow(), sqrt(), pi, e

Expressions available in all commands

BREAK for loops

IF may contain ()

BREAK loop



The screenshot shows a software interface for editing a sequencer file. At the top, there are four buttons: "Load Script", "Start Script", "Debug Script", and "Edit Script". Below these is a blue header bar with the text "Sequencer File". Underneath the header, it says "Filename: test.msl Comment:". The main area is a text editor containing the following code:

```
1 LOOP x, 10
2
3     IF ($x > 5)
4         BREAK
5     ENDIF
6
7     WAIT seconds, 3
8 ENDLOOP
9
10
```

Two blue arrows point from external text to the code. One arrow points from "IF may contain ()" to the opening parenthesis in line 3. The other arrow points from "BREAK loop" to the "BREAK" statement in line 4.

Array variables

```
LOOP i, 10  
  a[$i-1] = $i  
ENDLOOP
```

Loop variables run
from 1...n
Array indices run
from 0...n-1

```
ODBCREATE /Equipment/Test/Array, FLOAT, 10  
ODBSET /Equipment/Test/Array, a
```

New command

MSG to produce midas messages

The screenshot displays a control interface for a sequencer. At the top, there are three buttons: "Stop after current run", "Stop immediately", and "Pause". Below these are three main sections:

- Progress:** Shows "Loop: [10/10]" with a full purple progress bar and "Wait: [0/3 s]" with a green progress bar.
- Sequencer File:** Displays the filename "test.msl" and a list of commands:

```
1 LOOP x, 10
2 WAIT seconds, 3
3
4 MSG "Loop counter is $x"
5 ENDLLOOP
6
7
```

Line 4 is highlighted in blue, and a blue arrow points from it to the Messages section.
- Messages:** Shows a log of messages:

```
11:06:40.153 [Sequencer,INFO] Loop counter is 9
11:06:36.111 [Sequencer,INFO] Loop counter is 8
11:06:32.175 [Sequencer,INFO] Loop counter is 7
11:06:28.152 [Sequencer,INFO] Loop counter is 6
11:06:24.113 [Sequencer,INFO] Loop counter is 5
11:06:20.159 [Sequencer,INFO] Loop counter is 4
11:06:16.118 [Sequencer,INFO] Loop counter is 3
11:06:12.168 [Sequencer,INFO] Loop counter is 2
11:06:08.114 [Sequencer,INFO] Loop counter is 1
11:03:47.135 [Sequencer,TALK] Sequencer is finished.
...
```

ODBSAVE / ODBLOAD

- ▶ ODBLOAD used already to replace many ODBSET ...
- ▶ ODBSAVE /Equipment/.../Result file.odt
 - Can be used to save measurement results or snapshots of MPPC currents for example
 - Works with .odt, .xml and .json files

Full Example I-V curve

result.json

```
{
  "Output" : 50.9,
  "Current" : [
    5.6, 6.4, 7.5, 12.8, 54.3,
    78.2, 125.7, 175.5, 273.4, 389.5 ],
  "Voltage" : [
    50, 50.1, 50.2, 50.3, 50.4,
    50.5, 50.6, 50.7, 50.8, 50.9 ],
  "Input" : 389.5
}
```

The screenshot shows a control interface for a sequencer. At the top, there are three buttons: "Stop after current run", "Stop immediately", and "Pause". Below these are two progress bars: "Loop: [4/10]" with a blue bar and "Wait: [3/3 s]" with a green bar. The "Sequencer File" section shows the filename "test.msl" and a list of 24 lines of code. Line 14, "WAIT seconds, 3", is highlighted in green. The "Messages" section at the bottom shows a log of four messages, with the most recent one being "Measure 50.300000 volts". A blue arrow points from the "WAIT" line in the sequencer file to the "result.json" data on the left.

Stop after current run Stop immediately Pause

Progress

Loop: [4/10]

Wait: [3/3 s]

Sequencer File

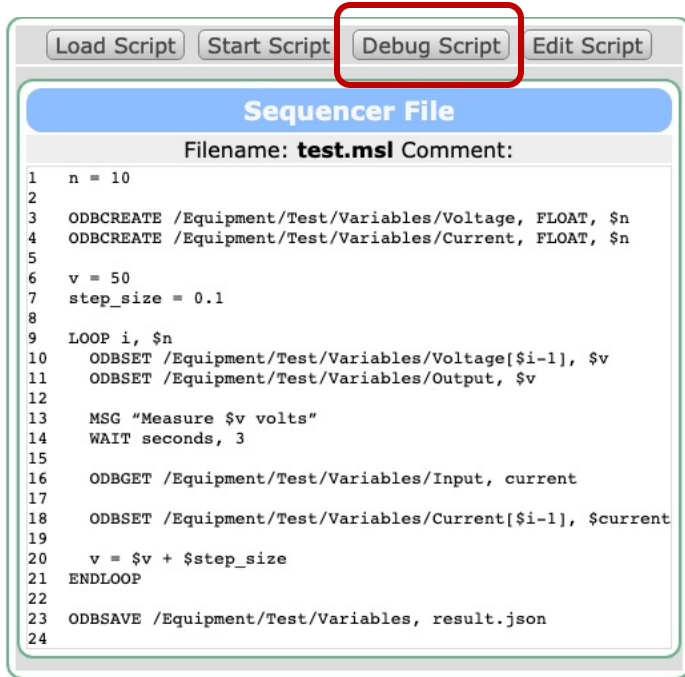
Filename: **test.msl** Comment:

```
1 n = 10
2
3 ODBC CREATE /Equipment/Test/Variables/Voltage, FLOAT, $n
4 ODBC CREATE /Equipment/Test/Variables/Current, FLOAT, $n
5
6 v = 50
7 step_size = 0.1
8
9 LOOP i, $n
10 ODB SET /Equipment/Test/Variables/Voltage[$i-1], $v
11 ODB SET /Equipment/Test/Variables/Output, $v
12
13 MSG "Measure $v volts"
14 WAIT seconds, 3
15
16 ODB GET /Equipment/Test/Variables/Input, current
17
18 ODB SET /Equipment/Test/Variables/Current[$i-1], $current
19
20 v = $v + $step_size
21 ENDOLOOP
22
23 ODB SAVE /Equipment/Test/Variables, result.json
24
```

Messages

```
11:51:53.195 [Sequencer,INFO] "Measure 50.300000 volts"
11:51:49.145 [Sequencer,INFO] "Measure 50.200000 volts"
11:51:45.193 [Sequencer,INFO] "Measure 50.100000 volts"
11:51:41.977 [Sequencer,INFO] "Measure 50 volts"
```

Debugging Scripts

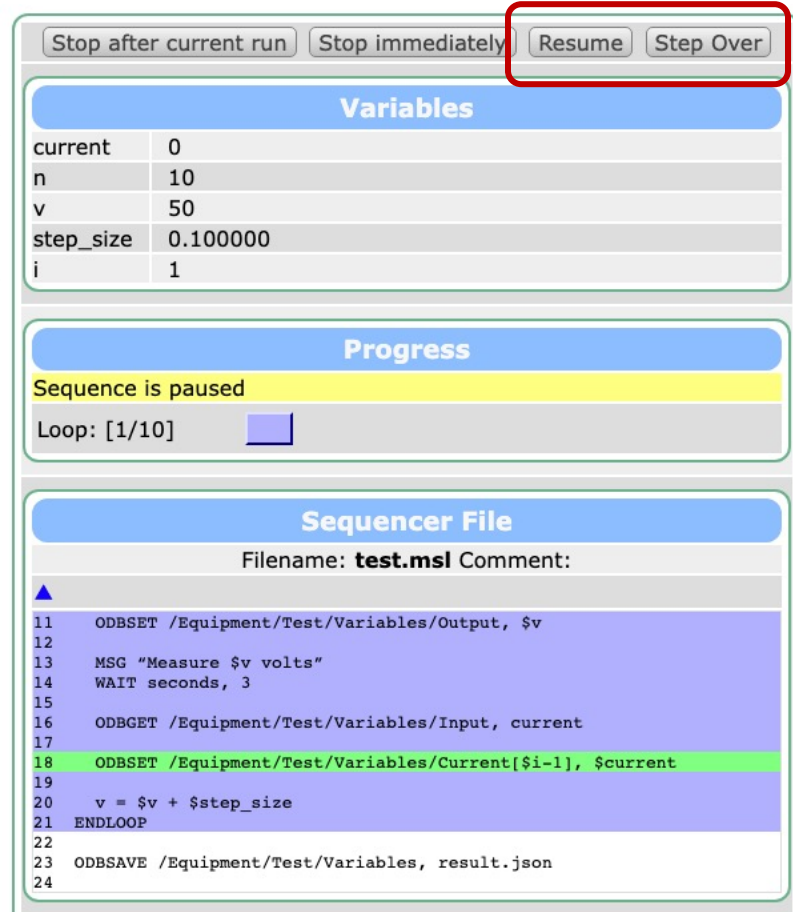


Load Script Start Script **Debug Script** Edit Script

Sequencer File
Filename: **test.msl** Comment:

```
1 n = 10
2
3 ODBCREATE /Equipment/Test/Variables/Voltage, FLOAT, $n
4 ODBCREATE /Equipment/Test/Variables/Current, FLOAT, $n
5
6 v = 50
7 step_size = 0.1
8
9 LOOP i, $n
10 ODBSET /Equipment/Test/Variables/Voltage[$i-1], $v
11 ODBSET /Equipment/Test/Variables/Output, $v
12
13 MSG "Measure $v volts"
14 WAIT seconds, 3
15
16 ODBGET /Equipment/Test/Variables/Input, current
17
18 ODBSET /Equipment/Test/Variables/Current[$i-1], $current
19
20 v = $v + $step_size
21 ENDLLOOP
22
23 ODBSAVE /Equipment/Test/Variables, result.json
24
```

Variables



Stop after current run Stop immediately **Resume** Step Over

Variables

current	0
n	10
v	50
step_size	0.100000
i	1

Progress

Sequence is paused

Loop: [1/10]

Sequencer File
Filename: **test.msl** Comment:

```
11 ODBSET /Equipment/Test/Variables/Output, $v
12
13 MSG "Measure $v volts"
14 WAIT seconds, 3
15
16 ODBGET /Equipment/Test/Variables/Input, current
17
18 ODBSET /Equipment/Test/Variables/Current[$i-1], $current
19
20 v = $v + $step_size
21 ENDLLOOP
22
23 ODBSAVE /Equipment/Test/Variables, result.json
24
```


Conclusions

- ▶ New functionality in MIDAS sequencer make it much more useful for us
- ▶ Can do “non-run-based” measurements
 - I-V curve
 - Beamline scan: change magnet, measure scaler
 - Results in ODB can be shown in custom page
- ▶ Debugging for easier code development
- ▶ Limitations:
 - Include files are not shown on web page (debugger sticks at “CALL”)
 - You might have to clear your browser cache to get new features
- ▶ If you are missing something, please let me know