



TABLE OF CONTENTS

Introduction	1
Macro for Repetitive (or Loop) Measurements	1
Input Data and Handling.....	2
Sample Applications.....	5
Example I.....	5
Example II.....	6
Individual Adaptations	6
Appendix - Source Code	7

Introduction

Using a macro commands that simulate keyboard action, measurements with ARTA can be automated to a large extent. This way, repetitive commands don't have to be carried out over and over again if the necessary parameters are predefined. For example, directivity measurements with a turntable can lead to a large number of necessary measurements, depending on the desired angle resolution. For such applications, a macro does not only improve general convenience – moreover, it helps to prevent slips of the pen that might occur due to monotone operational sequences. Beyond that, loop measurements come into consideration for all kinds of test series, e.g. with alternating microphone positions.

Using a AutoIT, a script language similar to Basic language, such a macro commands can be written easily. This application note provides user guide for the program that encapsulate macro commands. The program "Macro_for_ARTA_Loop_Measurement.exe" can be downloaded at www.fesb.hr/~mateljan/arta/support.htm. A source code of this program is given in the Appendix.

Macro for Repetitive (or Loop) Measurements

The present macro prompts the user to specify the desired behavior using a GUI (graphical user interface, image 1):



ARTA - APPLICATION NOTE

No 8: Repetitive Measurements with script language AutoIT

Macro for ARTA

Please specify the path to the ARTA.exe

Browse...

Measurement Setup

Waiting time between each measurement [in seconds]:

Filename prefix:

Filename suffix (also counts as step size):

Desired number of PIRs you want to record:

File Save Options

Save as PIR.

Save as ASCII.

Save as CSV.

Create an overlay for each measurement.

Okay - Start ARTA

Input Data and Handling

1. In the first input box, you have to specify the path to the ARTA.exe. This is required because the install path may differ depending on your windows language settings. Once you've entered the path and executed the macro, the information will be saved in



ART A - A P P L I C A T I O N N O T E

No 8: Repetitive Measurements with script language AutoIT

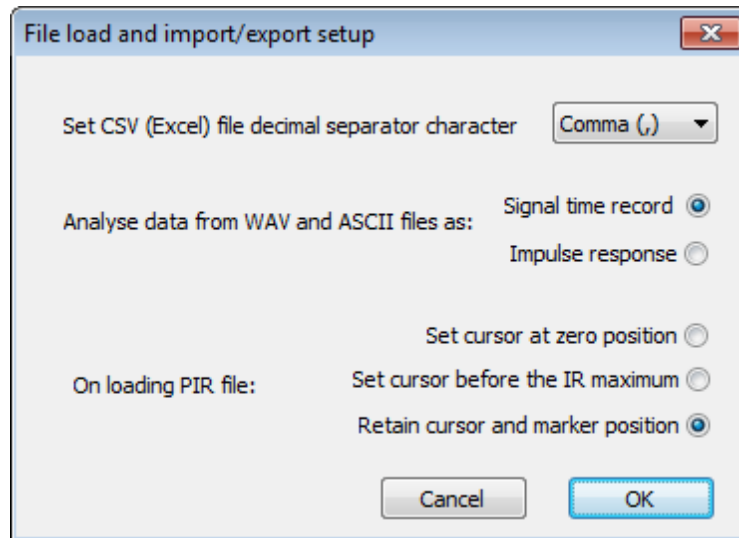
an ini-file to the macro directory, so you won't have to enter it again. This applies to the following input boxes as well.

2. In the "Measurement Setup" group, four input boxes have to be defined. For the first box, fill in your desired waiting time between each measurement. You must use a dot (.) as decimal separator character.
3. The file prefix can be chosen freely. If you are experiencing any problems with special characters, try to write them in curly brackets: "{example}".
4. The file suffix will be attached to the prefix you defined before. Moreover, the suffix functions as a step size parameter. The practical use of the step size parameter will be explained in the examples I and II (see page 4 - 5).
5. Enter a digit to specify how many measurements you want to record. Note that the starting value will always be 0. This value is not user-modifiable.
6. In a final step, the "File Save Options" group lets you choose between the ARTA file format for impulse responses (PIR), ASCII (TXT) or Excel files (CSV). All measurements will be saved in the folder that was used by ARTA the last time for saving. Please note that you can adjust the separator character for CSV-files under ARTA/ File/ Options (image 2), choosing either dots (.) or commas (,).



ARTA - APPLICATION NOTE

No 8: Repetitive Measurements with script language AutoIT



7. Moreover, you can choose to create an overlay for each measurement. If you are recording more than 20 measurements, then it is recommended to leave this box unchecked.
8. Once all the input boxes have been defined, you can start ARTA by clicking on the button at the bottom edge. Please consider that ARTA saves your last measurement signal setup, which this macro will use. So, if you get any errors, like: "FFT length exceeds number of samples", you have to adjust your sequence length/ sample rate settings. This macro will not check your measurement setup for consistency.
9. Now, the measurements will run in a loop. Once the specified number of measurements has been recorded, the script will end automatically. To run another loop measurement, close ARTA and run the macro again.
10. Please make sure that no other windows are opened while the macro is running because this could disrupt some of the commands that are being sent to ARTA. While the measurements are recorded, ARTA must not be put to the background!
11. Press the "pause/ break"-key to halt the macro. Press "pause/ break" again to continue.
12. Press the "ESC"-key to end the macro at any point.



ARTA - APPLICATION NOTE

No 8: Repetitive Measurements with script language AutoIT

Sample Applications

Example I

The aim of a measurement series shall be to record 50 microphone positions in a room. Because the microphone is positioned manually, we allow a pause of 15 seconds between each measurement. The filename shall be "position", while the suffix shall be 1 and counting upwards (each step +1).

For this example, the input data for the GUI has to be:

Macro for ARTA

Please specify the path to the ARTA.exe

C:\Program Files (x86)\ARTA Software\Arta.exe

Browse...

Measurement Setup

Waiting time between each measurement [in seconds]:

15

Filename prefix:

position

Filename suffix (also counts as step size):

1

Desired number of PIRs you want to record:

50

File Save Options

Save as PIR.

Save as ASCII.

Save as CSV.

Create an overlay for each measurement.

Okay - Start ARTA



ARTA - APPLICATION NOTE

No 8: Repetitive Measurements with script language AutoIT

Example II

The horizontal directivity of a loudspeaker shall be measured by rotating it about the frontal axis in increments of 5 degrees. In order to analyze the directivity pattern between 0 and 90 degrees, 19 measurements have to be recorded. We shall wait 2.5 seconds for the turntable to move the loudspeaker by 5 degrees. For this example, the input data should look like:

Macro for ARTA

Please specify the path to the ARTA.exe

C:\Program Files (x86)\ARTA Software\Arta.exe

Browse...

Measurement Setup

Waiting time between each measurement [in seconds]:

2.5

Filename prefix:

hor_deg+

Filename suffix (also counts as step size):

5

Desired number of PIRs you want to record:

18

File Save Options

Save as PIR.

Save as ASCII.

Save as CSV.

Create an overlay for each measurement.

Okay - Start ARTA

The ARTA filename format for use in directivity pattern files (.dpf) is "hor_deg+[degree]", "ver_deg-[degree]" etc.. Finally, the input box for the filename suffix also functions as a step size parameter. This means that the filename suffix will be multiplied with the number of measurements that were already performed in the loop (see source code, page 12).



ARTA - APPLICATION NOTE

No 8: Repetitive Measurements with script language AutoIT

Individual Adaptations

Of course, this macro will not meet the requirements of all applications that might be possible in combination with ARTA. This means (for example) sending commands to a motor driven turntable or moving a microphone via a robotic arm. At the moment, a macro for the Outline ET-2/ ST-2 is being worked on.

By means of the source code you will find attached, individual adaptations should be possible. The author is looking forward to hear from your variations.

Fabian Reimann (email: fabian.reimann@gmail.com)

Berlin, November 11, 2011, Rev. 1

Appendix - Source Code

To be used with AutoIT 3.3.6.1 and ARTA 1.7.

Working with Windows XP SP3 (32bit) and Windows 7 SP1 (32bit and 64bit).

(Not yet) working with Ubuntu LTS 10.4/ Wine 1.2.2.

```
; Macro for ARTA - loop measurements V. 0.9e  
; Fabian Reimann/ Berlin  
; 5/11/2011
```

```
Global $Paused
```

```
HotKeySet("{PAUSE}", "TogglePause")
```

```
HotKeySet("{ESC}", "Terminate")
```

```
$MacroExists = ProcessList("Macro_for_ARTA_Loop_Measurement.exe")
```

```
If $MacroExists [0] [0] = 2 Then
```

```
    MsgBox (16, "Macro for ARTA", "You can only run one macro at a time. This  
    process will now close.", 10)
```

```
    ProcessClose($MacroExists[2][1])
```

```
EndIf
```



ARTA - APPLICATION NOTE

No 8: Repetitive Measurements with script language AutoIT

```
#include <ButtonConstants.au3>
#include <GUIConstantsEx.au3>
#include <ProgressConstants.au3>
#include <StaticConstants.au3>
#include <WindowsConstants.au3>
#include <String.au3>
#include <WinApi.au3>
#include <GuiToolTip.au3>
#include <editconstants.au3>

#Region GUI

$form1_1 = GUICreate("Macro for ARTA", 378, 604, (@DesktopWidth - 378) / 2,
(@DesktopHeight - 604) / 2)

$Button1 = GUICtrlCreateButton("Browse...", 16, 64, 75, 25, $WS_GROUP)
$Button2 = GUICtrlCreateButton("Okay - Start ARTA", 64, 560, 243, 33,
$WS_GROUP)

$input1 = GUICtrlCreateInput("", 16, 32, 345, 21)
$input2 = GUICtrlCreateInput("", 16, 168, 177, 21)
$input3 = GUICtrlCreateInput("", 16, 224, 177, 21)
$input4 = GUICtrlCreateInput("", 16, 280, 177, 21)
$input5 = GUICtrlCreateInput("", 16, 336, 177, 21)

$Label1 = GUICtrlCreateLabel("", 16, 48, 4, 4)
$Label2 = GUICtrlCreateLabel("Waiting time between each measurement [in
seconds]:", 16, 144, 262, 17)
$Label3 = GUICtrlCreateLabel("Filename prefix:", 16, 200, 77, 17)
$Label4 = GUICtrlCreateLabel("Filename suffix (also counts as step size):",
16, 256, 197, 17)
$Label5 = GUICtrlCreateLabel("Desired number of PIRs you want to record:",
16, 312, 210, 17)

$Group1 = GUICtrlCreateGroup("Please specify the path to the ARTA.exe", 8, 8,
361, 97)

GUICtrlSetFont(-1, 8, 800, 0, "MS Sans Serif")
GUICtrlCreateGroup("", -99, -99, 1, 1)

$Group2 = GUICtrlCreateGroup("Measurement Setup", 8, 120, 361, 257)
GUICtrlSetFont(-1, 8, 800, 0, "MS Sans Serif")
GUICtrlCreateGroup("", -99, -99, 1, 1)

$Group3 = GUICtrlCreateGroup("File Save Options", 8, 392, 361, 153)
GUICtrlSetFont(-1, 8, 800, 0, "MS Sans Serif")
$Checkbox1 = GUICtrlCreateCheckbox(" Save as PIR.", 16, 416, 97, 17)
```




ARTA - APPLICATION NOTE

No 8: Repetitive Measurements with script language AutoIT

```
$Checkbox2 = GUICtrlCreateCheckbox(" Save as ASCII.", 16, 448, 97, 17)
$Checkbox3 = GUICtrlCreateCheckbox(" Save as CSV.", 16, 480, 97, 17)
$Checkbox4 = GUICtrlCreateCheckbox(" Create an overlay for each
measurement.", 16, 512, 297, 17)
GUICtrlCreateGroup("", -99, -99, 1, 1)
GUISetState(@SW_SHOW)
#EndRegion GUI

If FileExists (@ScriptDir & "\Macro for ARTA.ini") Then

    GUICtrlSetData($input1, IniRead(@ScriptDir & "\Macro for ARTA.ini",
"Section1", "Path", "No data!"))

    GUICtrlSetData($input2, IniRead(@ScriptDir & "\Macro for ARTA.ini",
"Section1", "Waiting Time", "No data!"))

    GUICtrlSetData($input3, IniRead(@ScriptDir & "\Macro for ARTA.ini",
"Section1", "Filename Prefix", "No data!"))

    GUICtrlSetData($input4, IniRead(@ScriptDir & "\Macro for ARTA.ini",
"Section1", "Filename Suffix", "No data!"))

    GUICtrlSetData($input5, IniRead(@ScriptDir & "\Macro for ARTA.ini",
"Section1", "Number of Measurements", "No data!"))

    $Checkbox1status = IniRead(@ScriptDir & "\Macro for ARTA.ini", "Section1",
"Checkbox1", "No data!")

    GuiCtrlSetState($CheckBox1, $Checkbox1status)

    $Checkbox2status = IniRead(@ScriptDir & "\Macro for ARTA.ini", "Section1",
"Checkbox2", "No data!")

    GuiCtrlSetState($CheckBox2, $Checkbox2status)

    $Checkbox3status = IniRead(@ScriptDir & "\Macro for ARTA.ini", "Section1",
"Checkbox3", "No data!")

    GuiCtrlSetState($CheckBox3, $Checkbox3status)

    $Checkbox4status = IniRead(@ScriptDir & "\Macro for ARTA.ini", "Section1",
"Checkbox4", "No data!")

    GuiCtrlSetState($CheckBox4, $Checkbox4status)
Else
EndIf

While 1

    $nMsg = GUIGetMsg()

    Switch $nMsg

        Case $GUI_EVENT_CLOSE
```



ARTA - APPLICATION NOTE

No 8: Repetitive Measurements with script language AutoIT

Exit

Case \$Button1

```
$dialog = FileOpenDialog("open", "C:\", "(ARTA.exe)")
```

```
GUICtrlSetData($input1, $dialog)
```

Case \$Button2

```
$time = GUICtrlRead($input2)
```

```
$timeforsleep = $time * 1000
```

```
$prefix_basic = GUICtrlRead($input3)
```

```
$prefix = StringReplace(StringReplace($prefix_basic, "+", "{+}"), "-",  
"{-}")
```

```
$suffix = GUICtrlRead($input4)
```

```
$NumberOfMeasurements = GUICtrlRead($input5)
```

```
$read = GUICtrlRead($input1)
```

```
$Check_Checkbox1 = ControlCommand("Macro for ARTA", "", 20, "IsChecked",  
"")
```

```
$Checkbox1status = GUICtrlRead($Checkbox1)
```

```
$Check_Checkbox2 = ControlCommand("Macro for ARTA", "", 21, "IsChecked",  
"")
```

```
$Checkbox2status = GUICtrlRead($Checkbox2)
```

```
$Check_Checkbox3 = ControlCommand("Macro for ARTA", "", 22, "IsChecked",  
"")
```

```
$Checkbox3status = GUICtrlRead($Checkbox3)
```

```
$Check_Checkbox4 = ControlCommand("Macro for ARTA", "", 23, "IsChecked",  
"")
```

```
$Checkbox4status = GUICtrlRead($Checkbox4)
```

```
SaveToIni()
```

```
GUIDelete()
```

ExitLoop

EndSwitch

```
CheckInput()
```

WEnd

Func CheckInput()

```
Local $string2 = GUICtrlRead($input2)
```



ARTA - APPLICATION NOTE

No 8: Repetitive Measurements with script language AutoIT

```
$string2mod = StringRegExpReplace($string2, "[^0-9\.]", "")

If $string2 <> $string2mod Then MsgBox ("64", "Macro for ARTA", "Please
enter a digit!" & @CRLF & "You can use a dot (.) as decimal separator
character.", 5)

GUISetData($input2, $string2mod)

Local $string4 = GUISetRead($input4)

$string4mod = StringRegExpReplace($string4, "[^0-9\.]", "")

If $string4 <> $string4mod Then MsgBox ("64", "Macro for ARTA", "Please
enter a digit!" & @CRLF & "You can use a dot (.) as decimal separator
character.", 5)

GUISetData($input4, $string4mod)

Local $string5 = GUISetRead($input5)

$string5mod = StringRegExpReplace($string5, "[^0-9]", "")

If $string5 <> $string5mod Then MsgBox ("64", "Macro for ARTA", "Please
enter a digit!", 5)

GUISetData($input5, $string5mod)

EndFunc

Func SaveToIni ()

    IniWrite(@ScriptDir & "\Macro for ARTA.ini", "Section1", "Path",
    GUISetRead($input1))

    IniWrite(@ScriptDir & "\Macro for ARTA.ini", "Section1", "Waiting Time",
    GUISetRead($input2))

    IniWrite(@ScriptDir & "\Macro for ARTA.ini", "Section1", "Filename Prefix",
    GUISetRead($input3))

    IniWrite(@ScriptDir & "\Macro for ARTA.ini", "Section1", "Filename Suffix",
    GUISetRead($input4))

    IniWrite(@ScriptDir & "\Macro for ARTA.ini", "Section1", "Number of
Measurements", GUISetRead($input5))

    IniWrite(@ScriptDir & "\Macro for ARTA.ini", "Section1", "Checkbox1",
    $Checkbox1status)

    IniWrite(@ScriptDir & "\Macro for ARTA.ini", "Section1", "Checkbox2",
    $Checkbox2status)

    IniWrite(@ScriptDir & "\Macro for ARTA.ini", "Section1", "Checkbox3",
    $Checkbox3status)

    IniWrite(@ScriptDir & "\Macro for ARTA.ini", "Section1", "Checkbox4",
    $Checkbox4status)

EndFunc
```



ARTA - APPLICATION NOTE

No 8: Repetitive Measurements with script language AutoIT

```
Run ($read)

WinWaitActive("Untitled - Arta")

WinMenuItem("Untitled - Arta","", "&Analysis", "Single-gated smoothed
&Frequency response / Spectrum")

WinWait("Smoothed frequency response (Untitled)", "Custom1", 5)

For $loop = 0 To $NumberOfMeasurements
    $stepvalue = $loop * $suffix

    Send("!f")
    Send("!r")

    WinWaitActive("Impulse response measurement")
    Send("!r")

    WinWaitClose("Impulse response measurement")

    ControlClick("Smoothed frequency response (Untitled)", "Custom1",
"Button1")

    if $Check_Checkbox1 = 1 Then
        Send("!f")
        Send("s")
        Sleep(300)
        Send($prefix & $stepvalue & ".pir")
        Send("{ENTER}")
        Sleep(300)
    EndIf

    if $Check_Checkbox2 = 1 then
        Send("!f")
        Send("e")
        Send("a")
        Sleep(300)
        Send($prefix & $stepvalue & ".txt")
        Send("{ENTER}")
        Sleep(300)
    EndIf
```



ARTA - APPLICATION NOTE

No 8: Repetitive Measurements with script language AutoIT

```
if $Check_Checkbox3 = 1 then
    Send ("!f")
    Send ("e")
    Send ("c")
    Sleep(300)
    Send($prefix & $stepvalue & ".csv")
    Send("{ENTER}")
    Sleep(300)
EndIf

if $Check_Checkbox4 = 1 then
    Send("!o")
    Send("a")
Endif

    Sleep($timeforsleep)
Next

Func TogglePause ()
    $Paused = Not $Paused
    While $Paused
        Sleep(100)
        ToolTip('Macro for ARTA is paused. Hit the button again to continue.',
0, 0)
    WEnd
    ToolTip("")
EndFunc

Func Terminate ()
    Exit 0
EndFunc
```