

Custom Math with VBScript

Create Custom Math Functions using simple scripts

WaveMaster™ series oscilloscopes incorporate the ability to create custom math and parameter functions based on the use of Visual Basic scripting. VBScripting is a fully supported and documented tool for automating operations within application programs. Microsoft provides extensive documentation and application notes on the language. In the WaveMaster scopes VBScripting is one of two tools used to create user specific math and parameters. The other supported program is MatLab.

WaveMaster currently supports two types of VBScript applications function scripts and parameter scripts. Function, or ‘wave scripts’ shown in the Math setup menu displayed in figure 1, take data from a trace and allows the user to create a new, separate, trace based on some manipulation of the source trace data. The parameter script uses data from the source trace to compute a single value parameter. Parameter scripts are found in the Measure menu definitions as ‘Param Script’

The advantage of VBScripts is that it allows user calculations to be entered into the scope processing chain and displayed on the scopes normal display. The language is commonly available and easy to learn. There are many books available to learn VBScript coding from. VBScript makes the WaveMaster scopes unique in the ability to perform custom math and parameter processing which is fully integrated into the oscilloscope .

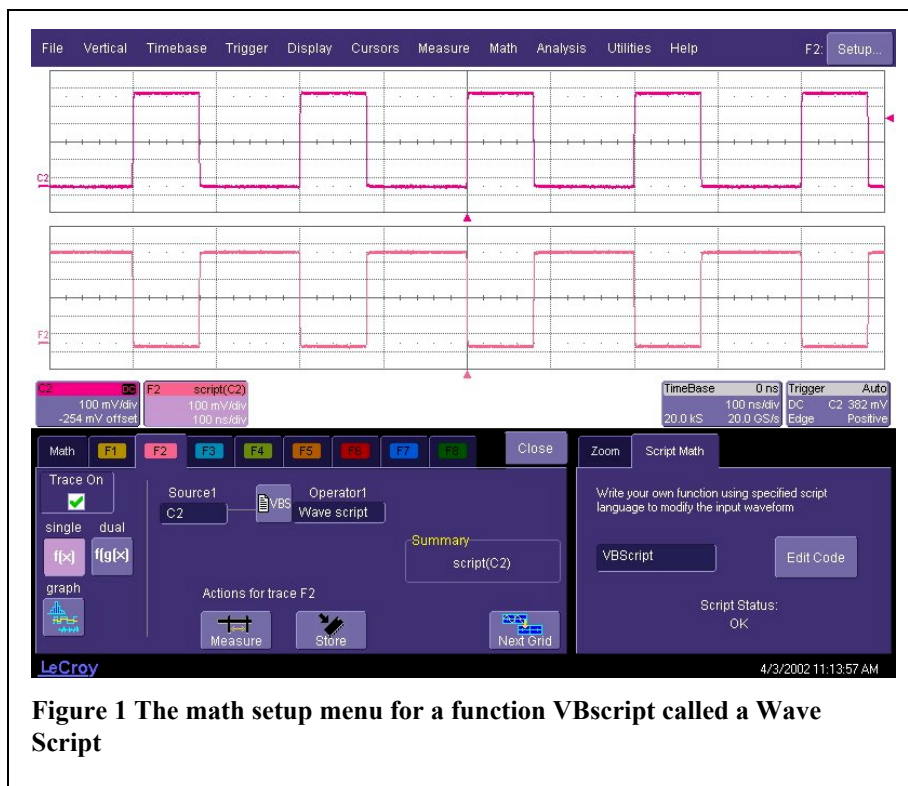


Figure 1 The math setup menu for a function VBScript called a Wave Script

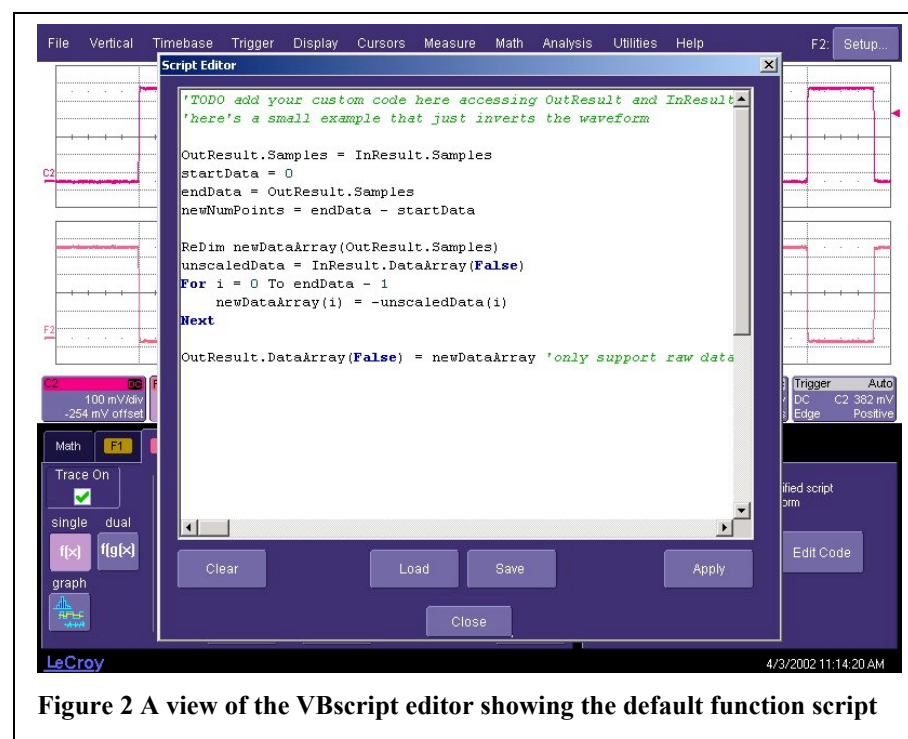


Figure 2 A view of the VBScript editor showing the default function script

Wave scripts can be created locally from the Math setup menu using the built-in script editor shown in figure 2. They can also be created offline using a text editor and loaded into the editor using the 'Load' button within the script editor. WaveMaster scopes include a default Wave script which is also shown in figure 2. The default script offers a simple example of how VBscripts work in the WaveMaster.

Figure 3 shows the default script in greater detail. The opening statements is prefaced with a single quote(') which indicates a comment. Any text after the quote is ignored by the VB interpreter.

VBscripts are example of object oriented programs. For those not familiar with object orientated programs, the notation is Object.Property or Object.Method. Properties define aspects of an object, for example, FontSize. Methods do things, for example, TurnBinary-IntoASCII(BinaryStuffIn, ASCIIStuffOut). In our example, the objects are InResult and OutResult. The property, '.Samples', describes the number of points in the data arrays. By deriving the number of samples directly from the input data the script can be used independent of the scopes time per division and memory length settings. The value of InResult.Samples is the total number of data in a trace. It is two more than the nominal value given on the screen. The first point DataArray(0), coincides with the left edge of the screen. If the trace length is nominally 500, the right edge of the screen coincides with DataArray(500), which is the 501st point. The last point, number 502, is just off the right of the screen, and is never seen. That is why the loop in the script runs to only endData - 1.

```
' TODO add your custom code here accessing OutResult
and InResult objects

' Here's a small example that just inverts the waveform.

OutResult.Samples = InResult.Samples

startData = 0

endData = OutResult.Samples

newNumPoints = endData - startData

ReDim newDataArray (OutResult.Samples)

unscaledData = InResult.DataArray (False)

    For i = 0 To endData - 1

        NewDataArray (i) = - unscaledData (i)

    Next

OutResult.DataArray (False) = newDataArray

' only support raw data
```

figure 3 The default Wave Script which inverts the input waveform

The property '.DataArray' contains the data values for the objects. The Boolean expression associated with the InResult.DataArray determines if the data array is scaled (Boolean value = True), that is read in units of the input trace such as Volts, or unscaled (Boolean value = False) and read as signed integer values in the range of +32,767 to -32768.

This example script reads in the data from the source trace and inverts its value. The resultant data object (OutResult) is returned to the scope for display.

VBscripting supports a range of control statements including For/Next, Do...Loop, If...Then, and

Select. These control structures allow iterative processes to be performed as shown in the example, where each value of the input data is replaced by its inverse.

Problems and errors are reported in the Wave Script menu at run time. Error messages normally provide a brief description of the problem and the line number at which it occurred.

Parameter scripts offer the same programming tools and differ only in that only the zero'th element of the output array is used.