PSBits / NoDLP / bin2wav.ps1

117 lines (96 loc) · 2.86 KB

Code Blame Raw

```
# PoC only. Filenames are hardcoded.

$msPerBit = 500 # must be divisible by 10
$srcData = Get-Content -Path C:\temp\test1.txt -Encoding Byte

$startStopBits = $true

[byte[]]$wavHeader = @()

#WAWRIFFHEADER
$wavHeader += (0x52, 0x49, 0x46, 0x46) #RIFF
$wavHeader += (0,0,0,0) #size1. to be filled later. 36+size2. Offset 0x04.
$wavHeader += (0x57, 0x41, 0x56, 0x45) #WAVE

#FORMATCHUNK
$wavHeader += (0x66, 0x6d, 0x74, 0x20) #fmt
$wavHeader += (0x10, 0x00, 0x00, 0x00) #chunk size
$wavHeader += (0x01, 0x00) #FormatTag = PCM
$wavHeader += (0x01, 0x00) #channels
$wavHeader += (0x44, 0xAC, 0x00, 0x00) #samples per second
$wavHeader += (0x44, 0xAC, 0x00, 0x00) #bytes per second
$wavHeader += (0x01, 0x00) #block align
$wavHeader += (0x08, 0x00) #bits per sample

#DATACHUNK
$wavHeader += (0x64, 0x61, 0x74, 0x61) #data
```

```
27      $wavHeader += (0x00, 0x00, 0x00, 0x00) # size2. Number of samples Offset 0x28
28
29      #size2
30      $samplesCount = 44.100 * $srcData.Count * $msPerBit * 8
31
32      if ($startStopBits)
33      {
34          $samplesCount += (44.100 * 2 * $msPerBit)
35      }
36
37      $wavHeader[0x28] = $samplesCount -band 0xFF
38      $wavHeader[0x29] = ($samplesCount -band 0xFF00) -shr 8
39      $wavHeader[0x2A] = ($samplesCount -band 0xFF0000) -shr 16
40      $wavHeader[0x2B] = ($samplesCount -band 0xFF000000) -shr 24
41
42      #totalsize
43      $wavHeader[0x04] = ($samplesCount + 36) -band 0xFF
44      $wavHeader[0x05] = (($samplesCount + 36) -band 0xFF00) -shr 8
45      $wavHeader[0x06] = (($samplesCount + 36) -band 0xFF0000) -shr 16
46      $wavHeader[0x07] = (($samplesCount + 36) -band 0xFF000000) -shr 24
47
48
49      $filename = 'C:\temp\test1.wav'
50      del $filename # for PoC
51      $fsw = new-object IO.FileStream($filename, [IO.FileMode]::CreateNew)
52      $writer = new-object IO.BinaryWriter($fsw)
53      $writer.Write($wavHeader)
54
55
56      $loFreq = 300
57      $hiFreq = 2000
58
59      $samplesPerBit = 44.1 * $msPerBit
60      $body = New-Object byte[] $samplesPerBit
61
62      # a bit of data for a start.
63      if ($startStopBits)
64      {
65          for ($k=0; $k -lt ($samplesPerBit); $k++)
66          {
67              $body[$k] = 0
68              if ((($k+1) % 4410) -eq 0)
69              {
70                  $body[$k] = 255
71              }
72          }
```

```
73          $writer.Write($body)
74      }
75
76      for ($i = 0; $i -lt $srcData.Count; $i++)
77      {
78          $srcByte = $srcData[$i]
79          for ($j = 0; $j -lt 8; $j++)
80          {
81              if (($srcByte -band (1 -shl (7 - $j))) -ne 0)
82              {
83                  $freq = $hiFreq
84              }
85              else
86              {
87                  $freq = $loFreq
88              }
89              for ($k=0; $k -lt ($samplesPerBit); $k++)
90              {
91                  $body[$k] = [byte](([System.Math]::Sin((2 * $k * [System.Math]::PI * $freq) / 44100) *
92              }
93              $writer.Write($body)
94          }
95      }
96
97      # a bit of data for a stop.
98      if ($startStopBits)
99      {
100         for ($k=0; $k -lt ($samplesPerBit); $k++)
101         {
102             $body[$k] = 0
103             if ((($k+1) % 4410) -eq 0)
104             {
105                 $body[$k] = 255
106             }
107         }
108         $writer.Write($body)
109     }
110
111
112     $fsw.Close()
113
114     # and now play it
115     # $PlayWav = New-Object System.Media.SoundPlayer
116     # $PlayWav.SoundLocation = $filename
117     # $PlayWav.PlaySync()
```