PLAY SAMPLE Command User Manual

Overview

The "PLAY SAMPLE" command plays back single-cycle waveforms stored in integer arrays with software-generated ADSR (Attack, Decay, Sustain, Release) amplitude envelopes. This enables synthesiser-style sound generation where the timbre is defined by the waveform shape and the dynamics are controlled by envelope parameters.
The waveform is looped continuously at the requested frequency using a phase accumulator, while the ADSR envelope shapes the amplitude over time. This is distinct from "PLAY ARRAY" which plays a pre-recorded buffer once at a given sample rate.
Note: This command is only available on RP2350-based PicoMite builds.

Syntax

Start Playback

```
PLAY SAMPLE left%(), right%(), freq, attack, decay, sustain, release
PLAY SAMPLE left%(), right%(), freq, attack, decay, sustain, release, interrupt
PLAY SAMPLE left%(), right%(), freqL, attackL, decayL, sustainL, releaseL, freqR, attackR, decayR, sust
PLAY SAMPLE left%(), right%(), freqL, attackL, decayL, sustainL, releaseL, freqR, attackR, decayR, sust
```

Trigger Release Phase

```
PLAY RELEASE
```

Stop Immediately

```
PLAY STOP
```

Pause / Resume

```
PLAY PAUSE
PLAY RESUME
```

Parameters

Parameter	Description
:---	:---
`left%()`	Integer array containing one cycle of the left-channel waveform. Va...
`right%()`	Integer array containing one cycle of the right-channel waveform. M...
`freq`	Playback frequency in Hz (10.0 to 48000.0). This is the pitch of th...
`attack`	Attack time in milliseconds (0 to 30000). Time for the envelope to ...
`decay`	Decay time in milliseconds (0 to 30000). Time for the envelope to f...
`sustain`	Sustain level as a percentage (0 to 100). The steady-state volume h...
`release`	Release time in milliseconds (0 to 30000). Time for the envelope to...
`freqR, attackR, d...	Optional right-channel pitch and ADSR parameters. When supplied, th...
`interrupt`	Optional. A label (subroutine) to call when the release phase compl...

If "freqR..releaseR" are omitted, both channels use "freq..release".

ADSR Envelope

The ADSR envelope controls how the volume of the note changes over time:

```
Volume
  ^
  |    /\
  |   /  \
  |  /    \___________
  | /      Sustain    \
  |/                  \
  +------+---+---------+----> Time
  Attack Decay         Release
         ^             ^
         |             |
    PLAY SAMPLE    PLAY RELEASE
```

1. **Attack**: When "PLAY SAMPLE" is executed, each channel volume ramps linearly from 0 to maximum over its attack time.
2. **Decay**: Each channel volume falls from maximum to its sustain level over its decay time.
3. **Sustain**: Each channel holds at its sustain level until "PLAY RELEASE" is issued.
4. **Release**: After "PLAY RELEASE", each channel fades from its sustain level to silence over its release time. When both channels complete release, playback stops and the optional interrupt is triggered.

Special Cases

- **Sustain = 0**: The note dies naturally after the decay phase without waiting for "PLAY RELEASE". This is useful for plucked or percussive sounds.
- **Attack = 0**: The note starts at full volume instantly.
- **Release = 0**: The note stops instantly when "PLAY RELEASE" is issued.
- **All times = 0, Sustain = 100**: Produces a continuous tone at full volume (equivalent to an organ stop).
- **Independent channel shaping**: Provide "freqR, attackR, decayR, sustainR, releaseR" to make left and right channels use different pitch and ADSR settings.

Waveform Arrays

Each array holds exactly one cycle of the waveform. The number of elements determines the resolution of the waveform shape - more elements give a smoother waveform with fewer harmonics. Typical sizes are 64 to 1024 elements.
The firmware uses a phase accumulator to step through the waveform table at the correct rate for the requested pitch, so the array size does not affect the output frequency - only the tonal quality.
Sample values should be in the range -32000 to +32000. Values outside this range may cause clipping.

Common Waveforms

Sine wave (pure tone, no harmonics):

```
  Const N = 256
  Dim Integer wave%(N - 1)
  For i% = 0 To N - 1
    wave%(i%) = Int(Sin(2 * Pi * i% / N) * 32000)
  Next i%
```

Square wave (odd harmonics, hollow sound):

```
  Const N = 256
  Dim Integer wave%(N - 1)
  For i% = 0 To N - 1
    If i% < N / 2 Then wave%(i%) = 32000 Else wave%(i%) = -32000
  Next i%
```

Sawtooth wave (all harmonics, bright/buzzy):

```
  Const N = 256
  Dim Integer wave%(N - 1)
  For i% = 0 To N - 1
    wave%(i%) = Int((i% / N * 2 - 1) * 32000)
  Next i%
```

Triangle wave (odd harmonics, softer than square):

```
  Const N = 256
  Dim Integer wave%(N - 1)
  For i% = 0 To N - 1
    Local Float t = i% / N
    If t < 0.5 Then
      wave%(i%) = Int((4 * t - 1) * 32000)
    Else
      wave%(i%) = Int((3 - 4 * t) * 32000)
    EndIf
  Next i%
```

Restarting Playback

"PLAY SAMPLE" can be called while a sample is already playing. The current note is stopped cleanly and the new note begins immediately. This allows playing melodies by issuing successive "PLAY SAMPLE" commands with different frequencies without needing to call "PLAY STOP" or "PLAY RELEASE" between notes.

Interaction with Other Audio Commands

- "PLAY SAMPLE" uses the same audio output hardware as "PLAY TONE", "PLAY SOUND", "PLAY WAV", "PLAY ARRAY", etc. Only one can be active at a time.
- Attempting to start "PLAY SAMPLE" while another audio mode is playing (other than a previous sample) will produce an error: ""Sound output in use for ..."".
- "PLAY VOLUME" affects sample playback - the ADSR envelope is applied on top of the global volume setting.
- "PLAY STOP" immediately halts sample playback at any point.

Examples

The current full demo program is included in "play_sample_demo.bas":

```
  ' ============================================================
  ' PLAY SAMPLE Demo - Independent Stereo ADSR Envelopes
  ' ============================================================
  ' This demo creates single-cycle waveforms and plays them
  ' using PLAY SAMPLE with the extended stereo syntax:
  '
  ' Basic (same envelope both channels):
  '   PLAY SAMPLE left%(), right%(), freq, A, D, S, R [,intr]
  '
  ' Extended (independent L/R frequency and ADSR):
  '   PLAY SAMPLE left%(), right%(),
  '     freq_L, A_L, D_L, S_L, R_L,
  '     freq_R, A_R, D_R, S_R, R_R [,intr]
  '
  ' The left channel carries the melody with a piano ADSR.
  ' The right channel carries a harmony with a string ADSR
  ' (slow attack, high sustain, long release).
  '
  ' PLAY RELEASE triggers both channels' release phase.
  ' PLAY STOP   stops immediately.
  ' ============================================================

  Option Base 0
  Const CYCLE_LEN = 256       ' samples per single cycle
  Const AMPLITUDE = 32000     ' peak amplitude (16-bit signed)

  Dim INTEGER left%(CYCLE_LEN - 1)
  Dim INTEGER right%(CYCLE_LEN - 1)

  ' --- Build a single-cycle sine wave ---
  For i% = 0 To CYCLE_LEN - 1
    left%(i%)  = Int(Sin(2 * Pi * i% / CYCLE_LEN) * AMPLITUDE)
    right%(i%) = left%(i%)
```

```
Next i%

' ============================================================
' Example 1: Basic mono playback (same ADSR both channels)
' ============================================================
Print "Example 1: Clean 440 Hz sine, 2 seconds..."
Print "  Attack=0  Decay=0  Sustain=100%  Release=0"
Play Sample left%(), right%(), 440, 0, 0, 100, 0
Pause 2000
Play Stop
Print "Done."
Print

' ============================================================
' Example 2: Piano-style envelope at middle C
' ============================================================
Print "Example 2: Piano tone at 262 Hz (middle C)..."
Print "  Attack=5ms  Decay=300ms  Sustain=20%  Release=500ms"
Play Sample left%(), right%(), 262, 5, 300, 20, 500
Pause 3000
Play Release
Pause 600
Print "Done."
Print

' ============================================================
' Example 3: Independent stereo - left piano, right string
'            Both play the same pitch (A4 = 440 Hz)
' ============================================================
' Demonstrates independent ADSR per channel at one pitch.
' Left  = Piano : fast attack, moderate decay, low sustain
' Right = String: slow attack, gentle decay, high sustain
Print "Example 3: Same pitch, different envelopes (stereo)..."
Print "  LEFT  (piano):  freq=440  A=5  D=300  S=20  R=500"
Print "  RIGHT (string): freq=440  A=150  D=200  S=75  R=800"
Play Sample left%(), right%(), 440, 5, 300, 20, 500, 440, 150, 200, 75, 800
Pause 3000
Play Release
Pause 900
Print "Done."
Print

' ============================================================
' Example 4: Stereo interval - left root, right fifth above
'            Left = C4 piano, Right = G4 string
' ============================================================
Print "Example 4: Stereo fifth - C4 piano (L) + G4 string (R)..."
Print "  LEFT  (piano):  freq=261.6  A=5  D=300  S=20  R=500"
Print "  RIGHT (string): freq=392.0  A=150  D=200  S=75  R=800"
Play Sample left%(), right%(), 261.6, 5, 300, 20, 500, 392.0, 150, 200, 75, 800
Pause 3000
Play Release
Pause 900
Print "Done."
Print

' ============================================================
' Example 5: Chromatic scale - piano left, string third right
' ============================================================
' The left channel plays a chromatic scale with a piano ADSR.
' The right channel plays a major third above with a string
' ADSR (the interval ratio is 5/4 = 1.2599).
Print "Example 5: Chromatic scale with stereo harmony..."
Restore scale_data
```

```
For note% = 1 To 13
  Read freq!
  harm! = freq! * 1.2599     ' major third above
  Print "  L="; freq!; "Hz  R="; harm!; "Hz"
  ' Left=piano(A=5,D=150,S=30,R=300) Right=string(A=120,D=150,S=75,R=500)
  Play Sample left%(), right%(), freq!, 5, 150, 30, 300, harm!, 120, 150, 75, 500
  Pause 500
  Play Release
  Pause 550
Next note%
Print "Scale complete."
Print

' ============================================================
' Example 6: Twinkle Twinkle Little Star
'   Left channel  = melody (piano ADSR)
'   Right channel = harmony a third above (string ADSR)
' ============================================================
' Piano ADSR:  Attack=5ms   Decay=150ms  Sustain=30%  Release=200ms
' String ADSR: Attack=150ms Decay=200ms  Sustain=75%  Release=800ms
'
' Melody in C major with harmony a diatonic third above:
'   C->E  D->F  E->G  F->A  G->B  A->C5
Print "Example 6: Twinkle Twinkle - piano melody (L) + string harmony (R)..."
Restore twinkle_harmony
For note% = 1 To 42
  Read mfreq!, hfreq!, dur%
  If mfreq! = 0 Then
    ' Rest
    Pause dur%
  Else
    Print "  melody="; mfreq!; "Hz  harmony="; hfreq!; "Hz  dur="; dur%; "ms"
    Play Sample left%(), right%(), mfreq!, 5, 150, 30, 200, hfreq!, 150, 200, 75, 800
    Pause dur%
    Play Release
    Pause 250
  EndIf
Next note%
Print "Tune complete."
End

done:
  ' Interrupt handler for end of release
  SamplePlaying% = 0
  IReturn

' Frequencies for one chromatic octave (A4 to A5)
scale_data:
Data 440.0, 466.2, 493.9, 523.3, 554.4, 587.3, 622.3
Data 659.3, 698.5, 740.0, 784.0, 830.6, 880.0

' Twinkle Twinkle Little Star with diatonic third harmony
' Each triple: melody freq, harmony freq, duration (ms)
' C4=261.6 D4=293.7 E4=329.6 F4=349.2 G4=392.0 A4=440.0 B4=493.9 C5=523.3
'
' Line 1: C C G G A A G-
' Harmony: E E B B C5 C5 B
twinkle_harmony:
Data 261.6, 329.6, 350, 261.6, 329.6, 350
Data 392.0, 493.9, 350, 392.0, 493.9, 350
Data 440.0, 523.3, 350, 440.0, 523.3, 350
Data 392.0, 493.9, 700
' Line 2: F F E E D D C-
' Harmony: A A G G F F E
```

```
  Data 349.2, 440.0, 350, 349.2, 440.0, 350
  Data 329.6, 392.0, 350, 329.6, 392.0, 350
  Data 293.7, 349.2, 350, 293.7, 349.2, 350
  Data 261.6, 329.6, 700
  ' Line 3: G G F F E E D-
  ' Harmony: B B A A G G F
  Data 392.0, 493.9, 350, 392.0, 493.9, 350
  Data 349.2, 440.0, 350, 349.2, 440.0, 350
  Data 329.6, 392.0, 350, 329.6, 392.0, 350
  Data 293.7, 349.2, 700
  ' Line 4: G G F F E E D-
  ' Harmony: B B A A G G F
  Data 392.0, 493.9, 350, 392.0, 493.9, 350
  Data 349.2, 440.0, 350, 349.2, 440.0, 350
  Data 329.6, 392.0, 350, 329.6, 392.0, 350
  Data 293.7, 349.2, 700
  ' Line 5: C C G G A A G-
  ' Harmony: E E B B C5 C5 B
  Data 261.6, 329.6, 350, 261.6, 329.6, 350
  Data 392.0, 493.9, 350, 392.0, 493.9, 350
  Data 440.0, 523.3, 350, 440.0, 523.3, 350
  Data 392.0, 493.9, 700
  ' Line 6: F F E E D D C-
  ' Harmony: A A G G F F E
  Data 349.2, 440.0, 350, 349.2, 440.0, 350
  Data 329.6, 392.0, 350, 329.6, 392.0, 350
  Data 293.7, 349.2, 350, 293.7, 349.2, 350
  Data 261.6, 329.6, 700
```

Technical Notes

- The output sample rate is fixed at 44,100 Hz (raised automatically if the requested frequency exceeds the Nyquist limit).
- The ADSR envelope uses 16.16 fixed-point arithmetic for smooth, click-free amplitude transitions.
- Each "PLAY SAMPLE" call allocates two 8 KB swing buffers for double-buffered output. These are freed when playback stops.
- The waveform arrays are read directly from MMBasic integer memory - they are not copied. Do not modify the arrays while playback is active.
- The "left%()" and "right%()" arrays must be one-dimensional integer arrays of the same size.

Error Messages

Error	Cause
:---	:---
`Sound output in us...	Another audio mode is already playing. Use `PLAY STOP` first.
`Invalid frequency ...	The `freq` parameter is outside the allowed range.
`Not playing a samp...	`PLAY RELEASE` was issued but no sample is currently playing.
`No program running...	An interrupt label was specified but the command was run from the...