

# Chapter 18 - The AHX Engine

AHX is a four-voice chip-tune music engine. It plays AHX module data from memory and generates synthetic square, saw, envelope, vibrato, and effect changes without sample data. The reader-facing path is simple: put AHX data in memory, set pointer and length, start playback, and poll status.

```
10 REM AHX MEMORY PLAYBACK
20 POKE32 &H000F0800,1
30 REM START SUBSONG 0 FROM MEMORY
40 AHX PLAY &H00100000,&H00004000,0
50 PRINT AHX STATUS
```

If valid AHX data is present at `&H00100000`, playback starts and `AHX STATUS` reports the busy bit.

Line 40 writes the pointer, length, subsong, and start command. Line 50 reads the status register once; it does not stop the tune.

## 18.1 Shape of the engine

Item	Value
Voices	4
Data	AHX or THX module data in memory
Subsongs	Selected by <code>AHX_SUBSONG</code>
Output	Mixed into the global audio output
Plus mode	Enhanced AHX processing at <code>AHX_PLUS_CTRL</code> bit 0
Status	Busy and error bits in <code>AHX_PLAY_STATUS</code>

AHX is a playback engine, not a register-level tone chip like SID or POKEY. There are no per-note BASIC commands in this chapter. Use the player registers or the `AHX PLAY` keyword.

## 18.2 Register block

Address	Name	Purpose
<code>\$F0B80</code>	<code>AHX_PLUS_CTRL</code>	Bit 0 enables AHX Plus
<code>\$F0B84</code>	<code>AHX_PLAY_PTR</code>	Start address of AHX data
<code>\$F0B88</code>	<code>AHX_PLAY_LEN</code>	Length in bytes
<code>\$F0B8C</code>	<code>AHX_PLAY_CTRL</code>	Write 1 start, 2 stop, 5 start loop
<code>\$F0B90</code>	<code>AHX_PLAY_STATUS</code>	Bit 0 busy, bit 1 error
<code>\$F0B91</code>	<code>AHX_SUBSONG</code>	Subsong number

AHX\_PLAY\_PTR, AHX\_PLAY\_LEN, and AHX\_PLAY\_CTRL are 32-bit registers. AHX\_SUBSONG is a byte. AHX\_PLUS\_CTRL can be read back as 0 or 1.

## 18.3 BASIC keywords

Form	Effect
AHX PLAY addr, len	Start playback from memory
AHX PLAY addr, len, subsong	Select a subsong, then start playback
AHX STOP	Stop playback
AHX PLUS ON	Enable AHX Plus processing
AHX PLUS OFF	Disable AHX Plus processing
AHX STATUS	Expression reading AHX_PLAY_STATUS

The equivalent raw register setup is:

```
10 REM AHX RAW START
20 POKE32 &H000F0800,1
30 REM POINTER, LENGTH, SUBSONG
40 POKE32 &H000F0B84,&H00100000
50 POKE32 &H000F0B88,&H00004000
60 POKE8 &H000F0B91,0
70 REM START, THEN READ STATUS
80 POKE32 &H000F0B8C,1
90 PRINT PEEK8(&H000F0B90)
```

Expected result: line 80 starts the player. Line 90 prints a status byte whose low bit is set while the player is busy, or bit 1 if the data cannot be read or parsed.

The raw form shows what AHX PLAY does for you. The pointer and length are 32-bit registers, the subsong is a byte register, and the control register starts playback when written with 1.

## 18.4 AHX Plus

AHX Plus follows the shared Plus rule from Chapter 11. AHX PLUS ON writes 1 to AHX\_PLUS\_CTRL at \$F0B80; AHX PLUS OFF writes 0. The AHX data and player registers stay the same. The AHX-specific difference is four-voice gain, stereo spread, oversampling, low-pass smoothing, drive, and room processing.

```

10 REM AHX PLUS TOGGLE
20 POKE32 &H000F0800,1
30 AHX PLAY &H00100000,&H00004000,0
40 REM LISTEN TO STANDARD AHX FIRST
50 FOR T=1 TO 3000
60 NEXT T
70 AHX PLUS ON
80 PRINT PEEK8(&H000F0B80)
90 REM NOW LISTEN TO AHX PLUS
100 FOR T=1 TO 3000
110 NEXT T
120 AHX PLUS OFF
130 PRINT PEEK8(&H000F0B80)

```

Expected result: line 80 prints 1; line 130 prints 0.

The song data and playback registers do not change when Plus is toggled. Only the output processing path changes, so playback continues while the two status prints confirm the control value.

## 18.5 Status and errors

AHX\_PLAY\_STATUS and AHX STATUS use these bits:

Bit	Meaning
0	Busy or playing
1	Error

The player sets the error bit if the memory pointer is not readable, the length is zero, the length is larger than the AHX limit, the address range wraps, or the data is not a valid AHX module.

```

10 REM AHX STATUS CHECK
20 REM START WITHOUT STOPPING IMMEDIATELY
30 AHX PLAY &H00100000,&H00004000,0
40 S=AHX STATUS
50 PRINT S
60 IF S AND 2 THEN PRINT "AHX ERROR"
70 IF S AND 1 THEN PRINT "AHX BUSY"

```

Line 40 samples the status register after the start command. Lines 60 and 70 decode the two useful bits. A good example should not call AHX STOP here, because that would prove only that stop works.

To stop playback later:

```

10 AHX STOP
20 PRINT AHX STATUS

```

## 18.6 Limits

The AHX player reads the module data from memory when playback starts. Changing the memory block after AHX PLAY does not change the already loaded song. AHX\_SUBSONG is an eight-bit value. AHX STOP writes control value 2 and clears the busy state.

For loading named music files from BASIC, use the media loader in Chapter 23.

The next chapter covers MOD playback, the sample-based four-channel music path.