Programming the Game Master Cartridge

Developing non-ROM Programs

Many developers have expressed a desire to make use of the audio synthesis capabilities of the Game Master Cartridge without distributing their programs on an actual Game Master Cartridge. While the Game Master Cartridge was designed with a game cartridge application in mind, use of the audio hardware by independent programs is completely possible through the use of a Tandy Multi-Pak Interface (MPI) or a compatible device. This is accomplished by configuring the MPI to boot from a different slot (e.g. slot 3 or 4 with a floppy drive controller or a CoCoSDC installed), then using the MPI's slot selection register at address $FF7F (decimal 65407) in order to make the Game Master Cartridge hardware available. (NOTE: Any subsequent use of the boot hardware will require a similar action to switch back to the boot slot.)

Suggestion

It is recommended that programs that use Game Master Cartridge hardware without being distributed on a Game Master Cartridge should expect (or even mandate) that the user should have a Game Master Cartridge (e.g. "Fahrfall: Master Edition") installed in Slot 1 of an MPI. This prevents having to query the user about what slot contains the Game Master Cartridge, enabling a more streamlined user experience. By establishing a "Slot 1" requirement for the Game Master Cartridge, such a program might include the snippet of code below in order to silently enable the audio hardware:

```
lda   $ff7f      Read current Slot Selection
anda  #$f0       Preserve CTS value, select Slot 1 for SCS
sta   $ff7f      Set new Slot Selection
```

Programs that are expected to exit cleanly may want to take precautions in order to save and restore the SCS value from when the program was started.

Using Emulators

Emulation can provide an attractive environment for software development. This is particularly true when developing code that uses "add-on" hardware, since not every developer will have that hardware physically present with them at any given time. Fortunately for Game Master Cartridge users, both Xroar and MAME provide emulation of the Game Master Cartridge

hardware. For simplicity, example command-line arguments will be provided to enable using those emulators to develop ROM images (e.g. “newgame.rom”) for use directly on the Game Master Cartridge.

Xroar

```
xroar -machine cocous -cart gmc -cart-rom newgame.rom
```

MAME[1][2]

```
mame coco2 -ext games_master -cart newgame.rom
```

Choosing specific machine configurations and various other options are left as an exercise to the reader, of course. Please consult the documentation for the chosen emulator.

Bank Switching

The CoCo Game Master Cartridge (GMC) provides a ROM bank switching capability that is identical to that used by the CoCo3 games RoboCop and Predator. That circuit was described by Greg L. Zumwalt in an article in the June 1990 edition of The Rainbow Magazine.

- http://archive.org/stream/rainbowmagazine-1990-06
- The ROM bank switching is handled identically in the CoCo Game Master Cartridge.
- The maximum ROM capacity is 64K.
- Each bank occupies almost 16K. The normal cartridge exceptions apply for addresses in the $FFXX range for all CoCo models and for addresses in the $FEXX range for the CoCo3.
- The bank select register on-board the GMC is addressed at $FF40. This register is write-only.

Sound Generation

In addition to the ROM bank switching capabilities, the GMC provides an SN76489AN sound generator. (General programming of the SN76489AN will not be covered in these notes.)

- http://en.wikipedia.org/wiki/Texas_Instruments_SN76489

---

[1] MAME users need to ensure that the "Becker port" is turned-off. This is due to addressing conflicts between the “Becker port” and the GMC hardware.

[2] In later versions of MAME (starting roughly around version 0237b), the “games_master” option was changed to “gmc” for unspecified reasons.

- Writes to the SN76489 require 32 cycles of the on-board oscillator (recommended 4 MHz) to complete successfully. Any attempt to write data to the chip too quickly will be lost, likely resulting in audible distortion or other audio problems.
- There is only a single address for writing to the SN76489. Individual registers internal to the chip are selected based upon the format of the data written to the single input register address.
- SN76489AN does not provide any readable status.
- The on-board oscillator frequency value is a factor in the values used to program the SN76489 tone generators. A change in the oscillator frequency must be reflected in data programmed to the chip or tones will be played at the incorrect pitch.
- The SN76489AN on-board the GMC is addressed at $FF41. This register is write-only.
- Tone generator register settings corresponding to the frequencies used for standard piano notes are available in the GMC SN76489AN Frequencies worksheet:

  https://drive.google.com/open?id=1Zwd5Xz-yf52WIs-L87meUyWL-jTPghXysL9al9-DnrM

System notes

- Address decoding for the GMC utilizes the “Spare Cartridge Select” (aka !SCS) signal from the CoCo’s expansion socket. This signal is normally used by the floppy disk controller on the CoCo, or more recently by a replacement for the floppy disk controller such as the CoCoSDC. Attempts to use the GMC along with such a card will require the use of a Multi-Pak Interface (MPI) or similar device which will need to be programmed with the information required to enable the GMC or other device as needed. For the MPI, this will involve programming the slot routing for the !SCS signal at address $FF7F, and will require knowing the slot numbers for the GMC and any other devices using !SCS.
- Audio output from the GMC’s SN76489AN is routed through the “SND” signal on the CoCo cartridge port. For audio to be heard, the CoCo’s internal analog mux will need to configured to route the external audio signal to the CoCo’s audio output. A BASIC example of the code needed for this is provided by lines 10-60 on page 9 of the Color Computer Speech-Sound Cartridge user’s manual. A short 6809 assembly example follows[3]:

```
lda   #$34       Enable MUX for sound from cartridge
sta   $ff01
lda   #$3f
sta   $ff03
lda   #$38       Enable sound through MUX
sta   $ff23
```

[3] An equivalent sequence is needed to enable audio input from any CoCo cartridge.

The sequence above could be expressed in Color BASIC as follows:

```
10 POKE &HFF01,52
20 POKE &HFF03,63
30 POKE &HFF23,56
```

- The tone generators on the SN76489AN default to being “on” at full volume when power is applied to the chip. It is advisable to first program the chip to minimize the volume on all output channels before enabling sound output from the GMC. A short 6809 assembly example follows:

```
lda   #$9f            Disable channel 0
sta   $ff41
nop
nop
lda   #$bf            Disable channel 1
sta   $ff41
nop
nop
lda   #$df            Disable channel 2
sta   $ff41
nop
nop
lda   #$ff            Disable channel 3
sta   $ff41
```

Alternatively, the following Color BASIC program can be used:

```
10 INPUT "GAME MASTER SLOT";S : REM OR LET S=<SLOT>
20 O=PEEK(&HFF7F)
30 N=(O AND 240)+S-1
40 POKE &HFF41,&H9F
50 POKE &HFF41,&HBF
60 POKE &HFF41,&HDF
70 POKE &HFF41,&HFF
80 POKE &HFF7F,O
```

NOTE: An AUTOEXEC.BAS program similar to the above might be used to silence the Game Master Cartridge for those that insist on leaving it mounted in an MPI along with other audio devices (e.g. the Speech/Sound Pak or the Orchestra-90). Setting the variable S in line 10 would eliminate the need for user interaction.

Playing/Composing Tunes

Mostly, you are “on your own” -- programming the SN76489 on the CoCo Game Master Cartridge is strictly an exercise for the user! With that said, I have a few suggestions that could be helpful.

- The combination of the SN76489 and a 4MHz oscillator yields a range of musical tones that does not include correct tunings for any notes lower than C3. However, the tunable range does extend upward to C8 and beyond.
- According to The Grand Staff, Clefs and Middle C, the treble clef (familiar to anyone that ever took a band class) covers the notes from C6 down to C4, while the bass clef covers the notes from C4 down to C2. Consequently, transcription of any music noted in the treble clef to the GMC should be trivial. Any music that restricts itself to the upper half of the bass clef should be trivial to transcribe for the GMC as well.
- Given the tuning ranges described above, essentially any music representable on the grand staff should be suitable for the GMC if the piece is simply transcribed to play an octave higher.
- The Texas Instrument SN76489 and compatible chips were used in a variety of the CoCo’s contemporary machines. These include the TI-99/4A, The SEGA Master System, the BBC Micro, the IBM PC Jr., and the Tandy 1000, among others. It is likely that any tunes composed for those systems and any tool used to compose tunes for those systems might be readily adaptable for use on a CoCo with the GMC.
- The VGM file format is compatible with the SN76489, the details of the format are reasonably well documented, and multiple archives of VGM formatted music exist. Just be sure to get SN76489-compatible VGMs, because the format covers other devices as well. Also, note the original system for the VGM data in case the frequency values need to be converted to account for the different SN76489 clocks.
- vgm-converter is a Python-based tool for converting SN76489-based VGM data in ways suitable for conversion between different machines. In particular it can transpose the tone generator data to account for different clock rates and it can correct the sampling rate to account for different register update timing during playback. Also, vgm-converter can produce output in a “raw” format (documented in the project’s README.md file) that simplifies processing and playback.
- DefleMask is a cross-platform tracker (runs on Windows, Mac OS X and Linux) for producing music for many soundchips and old school game-consoles/computers. It claims to support the SN76489, but you probably need to tell it to use a 4 MHz clock. (FWIW, the BBC Micro uses the same clock speed for the SN76489.)
- Ciaran Anscomb and Stewart Orchard from the Dragon community have come together to produce CyD-GMC, a tool to produce SID-like music using the audio hardware on the GMC. A git tree for CyD-GMC is available here: https://github.com/sorchard001/cyd-gmc

- Music composed for the YM-2149F (and equivalent chips like the AY-3-8910) may be found in chip music repositories encoded in YM format. Such files can be converted to SN76489-compatible VGM files using this project on Github: https://github.com/simondotm/ym2149f

Example Programs

gmcdemo.bas

This Color BASIC program demonstrates four simple sound effects using the SN76489 device on the Game Master Cartridge. The canned effects are a bell, a “phaser”, birds chirping, and an explosion.

gmcdemo.ccc

This is gmcdemo.bas packaged as a ROM image. See the “Color BASIC Program On A Cartridge?” blog post.

gmctool.bas

This Color BASIC program demonstrates how to calculate register values and how to program the registers of the SN76489 device on the Game Master Cartridge in order to make fundamental tones and noises. Volume control is demonstrated as well.

gmctool.ccc

This is gmctool.bas packaged as a ROM image. See the “Color BASIC Program On A Cartridge?” blog post.

cc76489.asm

This 6809 assembly language program is a cartridge ROM program that will play VGM files converted to the “raw” format by vgm-converter. The “raw” versions of multiple VGM files may be concatenated for extended audio playback.

Other Projects

GitHub hosts a number of projects related to the SN76489 chip used by the Game Master Cartridge. One or more of them might (or might not) be useful to you -- YMMV!

https://github.com/topics/sn76489

Recommended Videos

GMC Demo 1

https://www.youtube.com/watch?v=mGaaDpZ-M3c

GMC Demo 2

https://www.youtube.com/watch?v=TrxqOz7dJ5I

GMC Demo 3

https://www.youtube.com/watch?v=6A55D64fqu8

GMC Demo 4

https://www.youtube.com/watch?v=yzyDPGONsbw