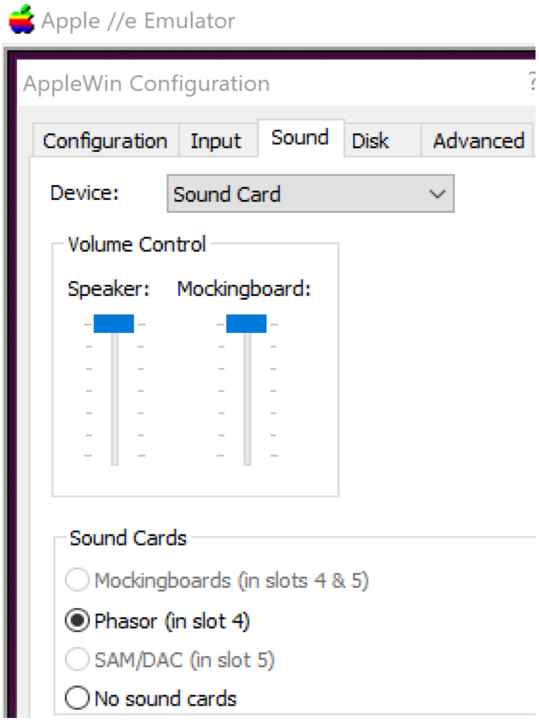
PLAYING THE PHASOR SOUND CARD

By Eric Rangell

I first found out about Kansasfest and the Apple II renaissance in June 2017 after moving out of an apartment and hooking up an Apple //c+ that had been sitting in my closet for years. I forgot how to boot it to 1MHz and Google connected me with our community (Thanks Quinn, and Peter N!). My experiences since then have felt like awakening from suspended animation. My long term memory was intact (although it took a few days for me to remember that '=' is the wildcard character for copying files in FID), and I began to dream of applications for vintage Apples in the modern world. During my teenage years I enjoyed musical applications on my Apple //e, and when I got a Mockingboard I was able to enter musical notation into programs that would play back multipart harmonies, such as Bank Street Music Writer. This was magical because I did not know what the music sounded like from the printed page, and had not yet learned to play piano (I took up the clarinet). I dove deeply into Mockingboard programming, and later MIDI programming for the Passport //e MIDI Interface (which can now be obtained from Bulgaria, thanks to Plamen!). These formative experiences enhanced my appreciation of classical, jazz, and popular musical styles of the 20th century.

I never owned an Applied Engineering Phasor Card, so when I first learned about it I was intrigued. It was essentially an upgraded Mockingboard that could play 12 voice music and had richer sound effects. When Henry Courbis of ReActiveMicro.com asked for volunteers to write test software for his Phasor V2 card, I signed up and found myself reverse engineering the sample code and poking away to create test programs that exercised the features of the card. In this article I would like to share my learnings so the community can easily come up to speed on what it takes to code new applications for this cool sound card.



Device: Sound Card

Sound Cards

○ Mockingboards (in slots 4 & 5)

● Phasor (in slot 4)

○ SAM/DAC (in slot 5)

○ No sound cards

Currently, Phasor emulation is available in the AppleWin emulator, in the Sound tab of the AppleWin Configuration panel. Select the radio button: **Phasor (in slot 4).** Slot 4 is the default slot for Mockingboard and Phasor sound cards, and many applications expect it there. However, the cards will work in any slot (except slot 3 for the //e and //gs) as long as the software allows you to select the slot for the card. Users of Mouse cards in slot 4 will need to experiment to find a configuration that works.

When you write your own programs, you should provide a configuration option to change the slot. There are ways to detect the slot where the card is installed, but that is beyond the scope of this article. Refer to sample programs on the Mockingboard and Phasor demonstration disks.

Disk images for the Phasor software can be downloaded from the ReActiveMicro Wiki at: https://wiki.reactivemicro.com/Phasor

Start your learning by playing each demo and noticing the sounds that are produced. You may find that some sounds are loud and are distorted. This is because the amplifier on the Phasor is much more powerful than the op-amps on the Mockingboard. You will need to adjust the volume controls on the sound card output and your speakers until the sounds levels are balanced. Start by sending a very low volume out of the card and amplifying it with your speakers (unamplified speakers will also work well). Then slowly increase the volume coming out of the card until it distorts, then lower it slightly. You can also play demonstration disks for the Mockingboard, and software which supports the Mockingboard. Take the time to enjoy the nostalgia of these sounds from the 1980s.

The Phasor has 4 sound generator chips, each of which has 3 sound channels which can play tones and/or noise.   Two of the chips are mapped to the left speaker and the other two to the right speaker. Fixed volume levels of 0-15 can be applied to any of the 12 voices independently, and your programs can vary those levels quickly in machine language to simulate envelopes and fades. To use the built-in sound envelopes (which are essentially square, triangle, and sawtooth waveforms), the envelope parameters on each chip control the 3 voices for that chip simultaneously. This gives you some flexibility to experiment with using different envelope shapes for each part in songs with 2 to 4 part harmonies. You can also experiment with creating different timbres by playing overtones of the primary note at different volumes. I may get into trouble for saying this, but the Phasor has the potential to create sounds that rival the Commodore 64 SID chip. Let the games begin!

If you have a Speech chip installed you can also run the Speech demos (Dr. Fay Zor is the Apple's Eliza), and play the original Crowther and Woods adventure game with speech (It came with the Mockingboard demo disks). If you have the time, you can even customize the speech rule tables to correct mispronunciations, and experiment with inflections and other speech parameters. The Phasor provides a simplified programming model for speech using the ampersand vector. Note: The Phasor manual mentions a compatibility mode for Echo speech synthesizer software, but neither Henry or I were able to figure out how to get it to work on the original AE card. Also, while some speech demos may work in AppleWin, it may have glitches or other issues with various programs.

Now let's experiment with programming sounds on the Phasor. Boot AppleWin and start up a disk image that lets you exit to Basic. Type CALL -151 to enter the monitor, then type the following lines:

```
C0CD
```

The monitor will return a value. You can ignore it.

```
C493:FF
C492:FF
C411:0  N C410:F N C410:C
C411:F4 N C410:E N C410:C
C411:1  N C410:F N C410:C
C411:0  N C410:E N C410:C
C411:7  N C410:F N C410:C
C411:38 N C410:E N C410:C
C411:8  N C410:F N C410:C
C411:A  N C410:E N C410:C
```

If you now hear a tone playing you have typed everything correctly. To stop the tone, enter:

```
C411:0 N C410:E N C410:C
```

Let’s play a little bit more:

```
C411:8  N C410:F N C410:C
C411:10 N C410:E N C410:C
C411:B  N C410:F N C410:C
C411:0  N C410:E N C410:C
C411:C  N C410:F N C410:C
C411:40 N C410:E N C410:C
C411:D  N C410:F N C410:C
C411:9  N C410:E N C410:C
```

You should now hear the tone play, and then decay in volume until it is silent.

```
C411:D  N C410:F N C410:C
C411:8  N C410:E N C410:C
```

You should now hear the tone play and decay repeatedly.

```
C411:C  N C410:F N C410:C
C411:8  N C410:E N C410:C
```

You should now hear the tone play and decay faster, repeatedly.

```
C411:6  N C410:F N C410:C
C411:F  N C410:E N C410:C
C411:7  N C410:F N C410:C
C411:7  N C410:E N C410:C
```

You should now hear noise instead of the tone, like a drumbeat.

```
C411:D  N C410:F N C410:C
C411:C  N C410:E N C410:C
```

You should now hear a train sound, where the noise repeatedly ramps up in a sawtooth pattern.

```
C411:C  N C410:F N C410:C
C411:80 N C410:E N C410:C
C411:D  N C410:F N C410:C
C411:E  N C410:E N C410:C
```

You should now hear the sound of the ocean.

To stop the sound (it will play a one-shot decay, then go silent):

```
C411:D  N C410:F N C410:C
C411:9  N C410:E N C410:C
```

Control-Reset also works as a panic button to stop all sounds, but you may need to re-enter the initialization commands when you want to play sound again:

```
C0CD
C493:FF
C492:FF
```

These 3 lines enable communications with the Phasor Card and initialize the sound chips. The addresses change based on the slot where the Phasor is installed. The address $C0CD is calculated as $C08D + $10*slot number. So slot 1 would be $C09D and slot 7 would be $C0FD. For the next 2 addresses, the second hex digit is the slot number, so slot 1 would be $C193 and slot 7 would be $C793.

In Applesoft Basic, this initialization can be done as follows:

```
REM SET SL=SLOT NUMBER WHERE PHASOR CARD IS INSTALLED
XX = PEEK (12*4096 + 16*SL + 8*16 + 13)
POKE 12*4096 + SL*256 + 9*16 + 3, 255
POKE 12*4096 + SL*256 + 9*16 + 2, 255
```

The Phasor uses the same procedures as the Mockingboard for setting values in the 6522 VIA registers, which control the behavior of the sound chips. The only differences are the addresses and bit patterns used. For more technical information about the sound parameters and programming procedures, please visit: **https://github.com/erangell/juicedgs-phasor**

Overall, the Phasor enables rich sound effects and music, which has potential applications for chipmusic, games, education, and whatever other applications you can imagine. ReActiveMicro redesigned the original card to reduce noise interference from the Apple motherboard, and simplified the hardware and connections, so this modern Apple sound card is an excellent product for our Vintage Apples.