Foenix Rising
A newsletter for developers, users, and enthusiasts of Foenix Retro Systems products
beta 1 - doc rev a.

What / why?

`sake`, based on *stand-alone-keys* (aka 'sak'), is a machine language program that scans and displays HID codes returned from MicroKernel events. `sake` is an 'extended' version of the original. (see Foenix Rising issue #13 for more on MicroKernel in general and a deep dive discussing mouse events, specifically).

The original 'keys' utility was written by Gadget as a programming example demonstrating how to leverage MicroKernel keyboard events. The original is built into every F256 and you can access it by entering DOS from SuperBASIC (typing "**/dos**" and then typing "**keys**" at the DOS prompt).

`sake` is further enhanced to support mouse and joystick events and features on-screen visuals representing meta-modifiers and hexadecimal key data. *If you are designing software that you hope will run consistently across F256 platforms, an understanding of keyboard codes is critical. This utility can help.*

Loading and execution

As opposed to the original sak program, `sake` is not packaged with a `.pgx` header. Instead, it is a set of binaries that can be loaded at prescribed addresses, then exited without disruption to your system.

BLOAD "{name}**",** ${addr}
CALL ${addr}

Or in future, load and call from a machine language monitor

The core code is < 512 bytes, however, the onscreen doc and loader code adds a kilobyte, totaling ~1,381 bytes (including the MicroKernel data struct footprint). Versions are named according to load/exec address at $2000, $4000, and $6000; distributed in a `.zip` file along with this one-page doc.

Keyboard and model differences

The F256 Jr. and F256K (depending on revision and add-on hardware), support 3 different keyboards but there are subtle differences to be aware of. This is especially important if your app uses meta and special keys (including function keys). The following list details differences between the PS/2 keyboard (applicable to the Jr. or when added to the F256K mouse port), the integrated F256K keyboard, and the 6522 interfaced 20-pin Commodore keyboard (enabled via DIP switch 6 and additional hardware on F256 Jr. machines):

	PS/2 (on a Jr. or plugged into an F256K mouse port)	F256K (integrated keyboard)	vintage Commodore keyboard plugged into a Jr's 20-pin header
PS/2 Caps Lock vs. the F256K LOCK vs. the CBM SHIFT LOCK	meta-key $08:$00:$80 (flagged*) **r**(aw):**a**(scii):**f**(lag)	does not trigger a MicroKernel event, but acts as a Caps Lock and illuminates the "LOCK" LED	actuates a hard SHIFT across all keys (special symbols and numeral keys)
RUN/STOP	(not applicable / no key)	standard key $BC:$03:$00	standard key $BC:$03:$00
right ALT	meta-key, raises flag bit	standard key $B8:$02:$00	(not applicable)
left ALT	meta-key, raises flag bit	same as PS/2, via "F" key	same as PS/2, via CBM key
F1 .. F8 (.. F12)	meta-keys $81 .. $8c :$00:$80	standard $81 .. $88 : (varied) : $00	standard $81 .. $88 : (varied) : $00
RESTORE	(not applicable / no key)	triggers NMI (no MicroKernel event)	triggers NMI (no MicroKernel event)

*the flag byte is either $00 (not-flagged) or $80 (flagged, aka negative) raised from meta-keys and in some cases, special-keys such as the PS/2 INSERT key

What else to know?

- Press <RETURN> to exit (cursor is turned back on)
- Code may be re-executed after exit by **CALL**'ing or `JSR`'ing to the load address
- The $4000 and $6000 versions are compatible with SuperBASIC's memory model
- Feedback / bugs? Please engage **EMwhite** on his channel on the Foenix Discord