Tech Info Library

PILOT: Controlling Memory Mapped I/O

Revised: 11/30/84
Security: Everyone

PILOT: Controlling Memory Mapped I/O

===

Many Apple II-compatible I/O devices are memory mapped; therefore, you can control them using Apple PILOT. The PILOT system variable %M is actually a memory map controlled by a subscript. The value of this subscript defines the location in the I/O device control space addressed by %M.

The I/O locations begin with address C080 (hex). This constant offset is added automatically to all %M locations. Variables %M(16) thru %M(127) address the device control space of slots 1 thru 7; access to slot 0 is not permitted. Variables %M(128) thru %M(1919) refer to the I/O select space of slots 1 thru 7. This space is typically used for ROM memory, but can be used in some cases for auxiliary device control information. For example, the PILOT command:

```
        C: %M(16) = 0
```

POKEs a zero into I/O location 16, which is the first available byte (relative location zero) in the device select space of slot 1. Similarly, the command:

```
        C: X = %M(20)
```

PEEKs into location 20 (relative location 4 in slot 1); it returns the result in variable X.

Generally variable %M may be used wherever the system variable %A is legal. It cannot be used to manipulate program memory from within PILOT lessons, but does provide some control of external devices.

The %M variable is a very powerful construct--I/O devices can be damaged by indiscriminate PEEKing and POKEing. Particularly avoid trying to manipulate any of the device controls for slot 6 where the disk controller is typically located. It is entirely possible to erase entire tracks of on your PILOT diskettes.

Apple Tech Notes

Tech Info Library Article Number:599