Apple II Miscellaneous
#8: Pascal 1.1 Firmware Protocol ID Bytes

Revised by: Matt Deatherage November 1988
Written by: Cameron Birse December 1986

This Technical Note documents the Pascal 1.1 Firmware Protocol ID bytes for Apple II peripheral cards and ports.

Background

Apple II Pascal 1.1 introduced a firmware protocol called, not surprisingly, the Pascal 1.1 Firmware Protocol. A card following this protocol could be identified by the following ID bytes, where n is the slot in which the card resides:

Address	Value	Definition
$Cn05	$38	ID byte (from Pascal 1.0)
$Cn07	$18	ID byte (from Pascal 1.0)
$Cn0B	$01	Generic signature of cards with Pascal 1.1 Protocol
$Cn0C	$ci	Device signature byte

$Cn0C was interpreted as two nibbles. The high-order nibble, c, was defined as the device signature. This signature was a pre-defined value determining what kind of device was connected (i.e., printer, modem, joystick, clock, etc.). The low-order nibble, i, was defined as a unique identifier, so you could tell one printer from another, for example.

Developer Technical Support no longer maintains a list of assignments for the i nibble in this protocol. Since, by definition, the Pascal 1.1 Protocol only has room for 16 uniquely identified devices of each signature, it is easy to see that the Apple II family has outgrown the definition.

Following is a table which lists the values of the Pascal 1.1 Firmware Protocol ID bytes for some Apple products which follow the protocol. Previous versions of this Note listed ID bytes for products which did not follow the protocol. Do **not** attempt to identify devices which do not follow the protocol by checking these ID bytes. This method will **not** work and should be avoided.

For example, trying to conclusively identify a 3.5” disk drive, SCSI hard drive, memory expansion card, or other SmartPort device using these ID bytes could be disastrous. For any SmartPort device, you should look for the ProDOS Block Device ID bytes ($Cn01 = $20, $Cn03

= $00, $Cn05 = $03), then look for the additional SmartPort ID byte ($Cn07 = $00). Once you have identified SmartPort, you should make a SmartPort `STATUS` call to determine the nature and types of connected devices. By this definition, ProDOS block devices and SmartPort devices cannot follow the Pascal 1.1 Firmware Protocol.

Pascal 1.1 Devices

	$Cn05	$Cn07	$Cn0B	$Cn0C
Apple II Peripheral Cards				
Super Serial Card (or port)	$38	$18	$01	$31
Apple 80 Column Card	$38	$18	$01	$88
Apple II Mouse Card	$38	$18	$01	$20
Apple IIc Ports				
1st version $FBBF = $FF				
Slot 1 (Serial Port)	$38	$18	$01	$31
Slot 2 (Serial Port)	$38	$18	$01	$31
Slot 3 (80 Columns)	$38	$18	$01	$88
Slot 4 (Mouse)	$38	$18	$01	$20
2nd version $FBBF = $00				
Slot 1 (Serial Port)	$38	$18	$01	$31
Slot 2 (Serial Port)	$38	$18	$01	$31
Slot 3 (80 Columns)	$38	$18	$01	$88
Slot 4 (Mouse)	$38	$18	$01	$20
Slot 7 (AppleTalk)	$38	$18	$01	$31
3rd version $FBBF = $03, 4th version $FBBF = $04, and 5th version $FBBF = $05				
Slot 1 (Serial Port)	$38	$18	$01	$31
Slot 2 (Serial Port)	$38	$18	$01	$31
Slot 3 (80 Columns)	$38	$18	$01	$88
Slot 7 (Mouse)	$38	$18	$01	$20
Apple IIGS Ports (ROM 1.0 and 2.0)				
Slot 1 (Serial Port)	$38	$18	$01	$31
Slot 2 (Serial Port)	$38	$18	$01	$31
Slot 3 (80 Columns)	$38	$18	$01	$88
Slot 4 (Mouse Port)	$38	$18	$01	$20
Slot 7 (AppleTalk)	$38	$18	$01	$31

ProDOS and SmartPort Devices

	$Cn01	$Cn03	$Cn05	$Cn07
Generic ProDOS Block Device	$20	$00	$03	$xx
SmartPort Device	$20	$00	$03	$00