

8042 - Keyboard Controller (AT,PS/2)

8042 Status Register (port 64h read)

```

|7|6|5|4|3|2|1|0| 8042 Status Register
| | | | | | | | `----- output register (60h) has data for system
| | | | | | | | `----- input register (60h/64h) has data for 8042
| | | | | | | | `----- system flag (set to 0 after power on reset)
| | | | | | | | `----- data in input register is command (1) or data (0)
| | | | | | | | `----- 1=keyboard enabled, 0=keyboard disabled (via switch)
| | | | | | | | `----- 1=transmit timeout (data transmit not complete)
| | | | | | | | `----- 1=receive timeout (data transmit not complete)
| | | | | | | | `----- 1=even parity rec'd, 0=odd parity rec'd (should be odd)

```

Port	Mode	Description
64h	read	8042 status register. Can be read at any time. See table above for more information.
64h	write	8042 command register. Writing this port sets Bit 3 of the status register to 1 and the byte is treated as a controller command. Devices attached to the 8042 should be disabled before issuing commands that return data since data in the output register will be overwritten.
60h	read	8042 output register (should only be read if Bit 0 of status port is set to 1)
60h	write	8042 data register. Data should only be written if Bit 1 of the status register is zero (register is empty). When this port is written Bit 3 of the status register is set to zero and the byte is treated as a data. The 8042 uses this byte if it's expecting data for a previous command, otherwise the data is written directly to the keyboard. See KEYBOARD COMMANDS for information on programming the actual keyboard hardware.

8042 Commands Related to PC Systems (Port 64h)

Command	Description
20	Read 8042 Command Byte: current 8042 command byte is placed in port 60h.
60	Write 8042 Command Byte: next data byte written to port 60h is placed in 8042 command register. Format:

```

|7|6|5|4|3|2|1|0| 8042 Command Byte

```

```

| | | | | | `---- 1=enable output register full interrupt
| | | | | | `----- should be 0
| | | | | | `----- 1=set status register system, 0=clear
| | | | | | `----- 1=override keyboard inhibit, 0=allow inhibit
| | | | | | `----- disable keyboard I/O by driving clock line low
| | | | | | `----- disable auxiliary device, drives clock line low
| | | | | | `----- IBM scancode translation 0=AT, 1=PC/XT
| | | | | | `----- reserved, should be 0

```

- A4 Password Installed Test: returned data can be read from port 60h; FA=password installed, F1=no password
- A5 Load Security: bytes written to port 60h will be read until a null (0) is found.
- A6 Enable Security: works only if a password is already loaded
- A7 Disable Auxiliary Interface: sets Bit 5 of command register stopping auxiliary I/O by driving the clock line low
- A8 Enable Auxiliary Interface: clears Bit 5 of command register
- A9 Auxiliary Interface Test: clock and data lines are tested; results placed at port 60h are listed below:

```

00 no error
01 keyboard clock line is stuck low
02 keyboard clock line is stuck high
03 keyboard data line is stuck low
04 keyboard data line is stuck high

```

- AA Self Test: diagnostic result placed at port 60h, 55h=OK
- AB Keyboard Interface Test: clock and data lines are tested; results placed at port 60h are listed above with command A9
- AC Diagnostic Dump: sends 16 bytes of 8042's RAM, current input port state, current output port state and 8042 program status word to port 60h in scan-code format.
- AD Disable Keyboard Interface: sets Bit 4 of command register stopping keyboard I/O by driving the clock line low
- AE Enable Keyboard Interface: clears Bit 4 of command register enabling keyboard interface.
- C0 Read Input Port: data is read from its input port (which is inaccessible to the data bus) and written to output register at port 60h; output register should be empty before call.

```

|7|6|5|4|3-0| 8042 Input Port
| | | | | `---- undefined
| | | | | `----- 1=enable 2nd 256K of motherboard RAM, 0=disable
| | | | | `----- 1=manufacturing jumper not installed, 0=installed
| | | | | `----- 1=primary display is MDA, 0=primary display is CGA
| | | | | `----- 1=keyboard not inhibited, 0=keyboard inhibited

```

- C1 Poll Input Port Low Bits: Bits 0-3 of port 1 placed in status Bits 4-7
- C2 Poll Input Port High Bits: Bits 4-7 of port 1 placed in status Bits 4-7
- D0 Read Output Port: data is read from 8042 output port (which is inaccessible to the data bus) and placed in output register; the output register should be empty. (see command D1 below)
- D1 Write Output Port: next byte written to port 60h is placed in

the 8042 output port (which is inaccessible to the data bus)

```

|7|6|5|4|3|2|1|0| 8042 Output Port
| | | | | | | `----- system reset line
| | | | | | | `----- gate A20
| | | | | | | `----- undefined
| | | | | | | `----- output buffer full
| | | | | | | `----- input buffer empty
| | | | | | | `----- keyboard clock (output)
| | | | | | | `----- keyboard data (output)

```

- D2 Write Keyboard Output Register: on PS/2 systems the next data byte written to port 60h input register is written to port 60h output register as if initiated by a device; invokes interrupt if enabled
- D3 Write Auxiliary Output Register: on PS/2 systems the next data byte written to port 60h input register is written to port 60h output register as if initiated by a device; invokes interrupt if enabled
- D4 Write Auxiliary Device: on PS/2 systems the next data byte written to input register a port at 60h is sent to the auxiliary device
- E0 Read Test Inputs: 8042 reads its T0 and T1 inputs; data is placed in output register; Bit 0 is T0, Bit 1 is T1:

```

|1|0| Test Input Port Bits
| | `----- keyboard clock
| | `----- keyboard data

```

- Fx Pulse Output Port: Bits 0-3 of the 8042 output port can be pulsed low for 6 µs; Bits 0-3 of command indicate which Bits should be pulsed; 0=pulse, 1=don't pulse; pulsing Bit 0 results in CPU reset since it is connected to system reset line.

- PC systems previous to the AT use the 8255 PPI as a keyboard controller and use the keyboard's internal 8048.
- the keyboard's internal controller buffers up to 16 bytes of make/break code information. This is common among all PC systems and shouldn't be confused with the (32 byte) keyboard buffer maintained by the BIOS.
- see [KEYBOARD COMMANDS](#) for information on programming the keyboards internal microprocessor