PS/2 Keyboard

From OSDev Wiki
(Redirected from PS2 Keyboard)

Contents

Overview

The PS/2 Keyboard is a device that talks to a PS/2 controller using serial communication. Ideally, each different type of PS/2 controller driver should provide some sort of standard/simple "send byte/receive byte" interface, and the PS/2 Keyboard driver would use this interface without caring about lower level details (like what type of PS/2 controller the device is plugged into).

The PS/2 Keyboard accepts commands and sends responses to those commands, and also sends scan codes indicating when a key was pressed or released.

Commands

A PS/2 Keyboard accepts many types of commands. A command is one byte. Some commands have data byte/s which must be sent after the command byte. The keyboard typically responds to a command by sending either an "ACK" (to acknowledge the command) or a "Resend" (to say something was wrong with the previous command) back.

The commands that a PS/2 Keyboard accepts are:

Command Byte	Data Byte/s	Meaning	Response
0xED	LED states: **Bit** | **Use** 0 | ScrollLock 1 | NumberLock 2 | CapsLock Note: Other bits may be used in international keyboards for other purposes (e.g. a Japanese keyboard might use bit 4 for a "Kana mode" LED).	Set LEDs	0xFA (ACK) or 0xFE (Resend)
0xEE	None	Echo (for diagnostic purposes, and useful for device removal detection)	0xEE (Echo) or 0xFE (Resend)
0xF0	Sub-command: **Value** | **Use** 0 | Get current scan code set 1 | Set scan code set 1	Get/set current scan code	0xFA (ACK) or 0xFE (Resend) if scan code is being set; 0xFA (ACK) then the scan code set number, or 0xFE (Resend) if you're getting the scancode

	2 Set scan code set 2 3 Set scan code set 3		
0xF2	None	Identify keyboard	0xFA (ACK) followed by none or more ID bytes (see [| "Detecting Device Types" (http://wiki.osdev.org/%228042%22_PS/2_Controller#Detecting_PS.2F2_Device_Types)])
0xF3	Typematic byte: **Bit/s** | **Meaning** 0 to 4 | Repeat rate (00000b = 30 Hz, ..., 11111b = 2 Hz) 5 to 6 | Delay before keys repeat (00b = 250 ms, 01b = 500 ms, 10b = 750 ms, 11b = 1000 ms) 7 | Must be zero	Set typematic rate and delay	0xFA (ACK) or 0xFE (Resend)
0xF4	None	Enable scanning (keyboard will send scan codes)	0xFA (ACK) or 0xFE (Resend)
0xF5	None	Disable scanning (keyboard won't send scan codes) Note: May also restore default parameters	0xFA (ACK) or 0xFE (Resend)
0xF6	None	Set default parameters	0xFA (ACK) or 0xFE (Resend)
0xF7	None	Set all keys to typematic/autorepeat only (scancode set 3 only)	0xFA (ACK) or 0xFE (Resend)
0xF8	None	Set all keys to make/release (scancode set 3 only)	0xFA (ACK) or 0xFE (Resend)
0xF9	None	Set all keys to make only (scancode set 3 only)	0xFA (ACK) or 0xFE (Resend)
0xFA	None	Set all keys to typematic/autorepeat/make/release (scancode set 3 only)	0xFA (ACK) or 0xFE (Resend)
0xFB	Scancode for key	Set specific key to typematic/autorepeat only (scancode set 3 only)	0xFA (ACK) or 0xFE (Resend)
0xFC	Scancode for key	Set specific key to make/release (scancode set 3 only)	0xFA (ACK) or 0xFE (Resend)
0xFD	Scancode for key	Set specific key to make only (scancode set 3 only)	0xFA (ACK) or 0xFE (Resend)
0xFE	None	Resend last byte	Previously sent byte or 0xFE (Resend)
0xFF	None	Reset and start self-test	0xAA (self-test passed), 0xFC or 0xFD (self test failed), or 0xFE (Resend)

Special Bytes

The keyboard sends bytes to the system. Some of these bytes have special meaning (e.g. responses from the commands above). The bytes the keyboard may send are:

Response Byte	Meaning
0x00	Key detection error or internal buffer overrun
0xAA	Self test passed (sent after "0xFF (reset)" command or keyboard power up)
0xEE	Response to "0xEE (echo)" command
0xFA	Command acknowledged (ACK)
0xFC and 0xFD	Self test failed (sent after "0xFF (reset)" command or keyboard power up)

0xFE	Resend (keyboard wants controller to repeat last command it sent)
0xFF	Key detection error or internal buffer overrun

All other bytes sent by the keyboard are scan codes, where interpretation depends on the currently selected scan code set.

Driver Model

Command Queue and State Machine

Commands must be sent one at a time (e.g. if your driver is interrupt driven, you can't start sending a command within the IRQ handler because code outside the IRQ handler may be in the middle of sending a command). The command isn't completed until you've received an ACK for it. For example, if you send a command and the keyboard responds with "0xFE (resend)" then you have to send the command again (possibly limited to 3 retries before you give up and assume the keyboard doesn't support the command you're sending or there's been a hardware failure). Finally, sometimes you want to send several commands at once. For example, you might have a "reinitialise()" function that sets the scan code set, sets the typematic byte, sets the LEDs and enables scanning.

The simplest way to achieve this is for the driver to maintain a queue of commands. When you add a command to the queue, if the queue is empty you start sending the command; otherwise you append the command to the queue. When you receive an "0xFA (ACK)" from the keyboard you discard the command at the head of the queue and start sending the next command in the queue (if any). If you receive an "0xFE (Resend)" from the keyboard you can resend the command at the head of the queue.

The remainder of the driver should be a kind of state machine. The state machine moves into a different state when some commands are successfully completed, and when various bytes are received from the keyboard. For example, the driver might be in a default state and receive a break code that puts it into a "waiting for scan code after receiving break code" state. Then it might receive the first byte of a multi-byte scan code and shift to a "waiting for second byte of scan code after receiving break code" state. Finally it might receive the second/last byte of the scan code and then switch back to the default state.

Scan Code Sets, Scan Codes and Key Codes

A scan code set is a set of codes that determine when a key is pressed or repeated, or released. There are 3 different sets of scan codes. The oldest is "scan code set 1", the default is "scan code set 2", and there is a newer (more complex) "scan code set 3". *Note: Normally on PC compatible systems the keyboard itself uses scan code set 2 and the keyboard controller translates this into scan code set 1 for compatibility. See "8042"_PS/2_Controller for more information about this translation.*

Modern keyboards should support all three scan code sets, however some don't. Scan code set 2 (the default) is the only scan code set that is guaranteed to be supported. In theory (I haven't tried it) it should be possible to attempt to set scan code set 1 or scan code set 3, and then ask the keyboard which scan code it is currently using and see if it actually is using the requested scan code set. In this way it may be possible to determine which scan code sets the keyboard does support.

Scan codes themselves are sequences of one or more bytes. In some cases the sequence can be as many as 6 bytes (e.g. the "print screen" key in scan code set 1 generates the sequence 0xE1, 0x1D, 0x45, 0xE1, 0x9D, 0xC5 when pressed). This situation isn't really ideal. In general (for later processing) you want to convert these "one or more byte sequences" into a single integer that uniquely identifies a specific key, that can be used effectively in things like lookup tables (without having sparsely used "many GiB" lookup tables).

There is no standard for "key codes" - it's something you have to make up or invent for your OS. I personally like the idea of having an 8-bit key code where the highest 3 bits determine which row on the keyboard and the lowest 5 bits determine which column (essentially, the keyboard is treated as a grid of up to 8 rows and up to 32 columns of keys). Regardless of what you choose to use for your key codes, it should be something that is used by all keyboard drivers (including USB Keyboards) and could possibly also be used for other input devices (e.g. left mouse button might be treated as "key code 0xF1").

Basically, when the keyboard driver's state machine knows it has received a complete scan code, the next step is to convert the "one or more byte" scan code into a key code.

Key Codes, Key States and Key Mappings

Once you've got key codes, then next step is to keep track of which keys are currently being pressed. Imagine a computer game that uses the "WASD" keys for player movement - when the 'A' key is being pressed the player rotates clockwise. How does the game know if the 'A' key is currently being pressed? For this you'd want an array of flags, where each flag corresponds to a key code. There is a hidden bonus here - the keyboard driver itself can use the same "array of flags" to determine if a shift key, control key, alt key, etc is down, which can be quite useful when trying to convert the key code into an actual ASCII character or Unicode code point. For example, if the user presses the 'a' key then it might correspond to 'a' or 'A' (depending on capslock state and whether or not a shift key is being held down at the time) or might not correspond to a valid character at all (e.g. "control-a" or "alt-a").

Also note that (for example) a "WASD" game doesn't care if the keys are 'W', 'A', 'S' and 'D'. The game wants to know about keys in a specific "T-shaped" pattern on the left of the keyboard. If the keyboard happens to be something different, then the keys in the same location may be completely different (e.g. they would be '<', 'A', 'O' and 'E' keys on a Dvorak keyboard). This helps to explain my preference of having an 8-bit key code where the highest 3 bits determine which row on the keyboard and the lowest 5 bits determine which column (it's easy for a game to ask about the state of the third key on the left of the third row).

Once you're able to keep track of which keys are currently being pressed, the next step is to (attempt to) convert the key into an ASCII character or Unicode code point. At this point you need to know what type of keyboard the user has - is it "US QWERTY", or "French AZERTY", some form of Dvorak, or perhaps it's Arabic. To handle many different keyboard layouts, the keyboard driver needs to use tables to convert key codes into ASCII characters or Unicode code points; so that you only need to load a different "Key Mapping" table to support different keyboard layouts.

However, it's not quite that simple. Different keyboard layouts can have different meta keys, different status LEDs, etc. The Key Mapping table has to sort these differences out too. This is why you don't want to detect if the keyboard LEDs have changed earlier, but want to send the "set LEDs" command (if necessary) *after* you've found the entry for the key code in your key map table.

The final step of processing is to combine all relevant information into some sort of "keypress packet" structure, and send it to whomever (e.g. GUI). The entire "keypress packet" might include the following:

- Unicode code point (if applicable)
- Key code
- Pressed/released flag
- Various other key states (shift, alt, control, etc)
- Various "toggle" states (CapsLock, ScrollLock, NumberLock, etc)

Scan Code Sets

As there are 3 different scan code sets, there are 3 different tables (one for each scan code set). Some of the scan codes correspond to extra keys that have been added over time and have become "relatively standard". To help keep track scan codes have been categorized and tagged in the tables below. The tags used are:

Tag	Meaning
(keypad)	A key that is on the numerics keypad (typically found on the right hand side of the keyboard).
(ACPI)	A key that is part of the "ACPI" group of keys (typically found near the top of the keyboard). A lot of modern keyboards don't actually have these keys (if I remember right, they were fashionable in the late 1990's but have become less common since). Note: Don't let the name fool you - these keys have nothing to do with ACPI at all and do behave like any other normal key (but could be useful for an OS that supports power management).
(multimedia)	A key that is part of the multimedia group of keys (typically found near the top of the keyboard). A lot of modern keyboards do have at least some of these keys. Some of these keys are intended to be used for media players (volume control, play/pause, next track, previous track, etc), some are intended for web browsing (previous web page, next web page, refresh, favourites/bookmarks, etc), and and some are intended for launching applications (e.g. starting an email client, starting a calculator, opening "my computer", etc).

Scan Code Set 1

The following table shows which scan codes correspond to which keys when using scan code set 1 (for a "US QWERTY" keyboard only):

Scan code	Key	Scan code	Key	Scan code	Key	Scan code	Key
		0x01	escape pressed	0x02	1 pressed	0x03	2 pressed
0x04	3 pressed	0x05	4 pressed	0x06	5 pressed	0x07	6 pressed
0x08	7 pressed	0x09	8 pressed	0x0A	9 pressed	0x0B	0 (zero) pressed
0x0C	- pressed	0x0D	= pressed	0x0E	backspace pressed	0x0F	tab pressed
0x10	Q pressed	0x11	W pressed	0x12	E pressed	0x13	R pressed
0x14	T pressed	0x15	Y pressed	0x16	U pressed	0x17	I pressed
0x18	O pressed	0x19	P pressed	0x1A	[pressed	0x1B	] pressed
0x1C	enter pressed	0x1D	left control pressed	0x1E	A pressed	0x1F	S pressed
0x20	D pressed	0x21	F pressed	0x22	G pressed	0x23	H pressed
0x24	J pressed	0x25	K pressed	0x26	L pressed	0x27	; pressed
0x28	' (single quote) pressed	0x29	` (back tick) pressed	0x2A	left shift pressed	0x2B	\ pressed
0x2C	Z pressed	0x2D	X pressed	0x2E	C pressed	0x2F	V pressed
0x30	B pressed	0x31	N pressed	0x32	M pressed	0x33	, pressed
0x34	. pressed	0x35	/ pressed	0x36	right shift pressed	0x37	(keypad) * pressed
0x38	left alt pressed	0x39	space pressed	0x3A	CapsLock pressed	0x3B	F1 pressed
0x3C	F2 pressed	0x3D	F3 pressed	0x3E	F4 pressed	0x3F	F5 pressed
0x40	F6 pressed	0x41	F7 pressed	0x42	F8 pressed	0x43	F9 pressed
0x44	F10 pressed	0x45	NumberLock pressed	0x46	ScrollLock pressed	0x47	(keypad) 7 pressed
0x48	(keypad) 8 pressed	0x49	(keypad) 9 pressed	0x4A	(keypad) - pressed	0x4B	(keypad) 4 pressed
0x4C	(keypad) 5 pressed	0x4D	(keypad) 6 pressed	0x4E	(keypad) + pressed	0x4F	(keypad) 1 pressed
0x50	(keypad) 2 pressed	0x51	(keypad) 3 pressed	0x52	(keypad) 0 pressed	0x53	(keypad) . pressed
						0x57	F11 pressed
0x58	F12 pressed						
		0x81	escape released	0x82	1 released	0x83	2 released
0x84	3 released	0x85	4 released	0x86	5 released	0x87	6 released
0x88	7 released	0x89	8 released	0x8A	9 released	0x8B	0 (zero) released

0x8C	- released	0x8D	= released	0x8E	backspace released	0x8F	tab released
0x90	Q released	0x91	W released	0x92	E released	0x93	R released
0x94	T released	0x95	Y released	0x96	U released	0x97	I released
0x98	O released	0x99	P released	0x9A	[released	0x9B	] released
0x9C	enter released	0x9D	left control released	0x9E	A released	0x9F	S released
0xA0	D released	0xA1	F released	0xA2	G released	0xA3	H released
0xA4	J released	0xA5	K released	0xA6	L released	0xA7	; released
0xA8	' (single quote) released	0xA9	` (back tick) released	0xAA	left shift released	0xAB	\ released
0xAC	Z released	0xAD	X released	0xAE	C released	0xAF	V released
0xB0	B released	0xB1	N released	0xB2	M released	0xB3	, released
0xB4	. released	0xB5	/ released	0xB6	right shift released	0xB7	(keypad) * released
0xB8	left alt released	0xB9	space released	0xBA	CapsLock released	0xBB	F1 released
0xBC	F2 released	0xBD	F3 released	0xBE	F4 released	0xBF	F5 released
0xC0	F6 released	0xC1	F7 released	0xC2	F8 released	0xC3	F9 released
0xC4	F10 released	0xC5	NumberLock released	0xC6	ScrollLock released	0xC7	(keypad) 7 released
0xC8	(keypad) 8 released	0xC9	(keypad) 9 released	0xCA	(keypad) - released	0xCB	(keypad) 4 released
0xCC	(keypad) 5 released	0xCD	(keypad) 6 released	0xCE	(keypad) + released	0xCF	(keypad) 1 released
0xD0	(keypad) 2 released	0xD1	(keypad) 3 released	0xD2	(keypad) 0 released	0xD3	(keypad) . released
						0xD7	F11 released
0xD8	F12 released						
0xE0, 0x1C	(keypad) enter pressed	0xE0, 0x1D	right control pressed				
		0xE0, 0x35	(keypad) / pressed				
0xE0, 0x38	right alt (or altGr) pressed						
						0xE0, 0x47	home pressed
0xE0, 0x48	cursor up pressed	0xE0, 0x49	page up pressed			0xE0, 0x4B	cursor left pressed
		0xE0, 0x4D	cursor right pressed			0xE0, 0x4F	end pressed
0xE0, 0x50	cursor down pressed	0xE0, 0x51	page down pressed	0xE0, 0x52	insert pressed	0xE0, 0x53	delete pressed
						0xE0, 0x5B	left GUI pressed
0xE0, 0x5C	right GUI pressed	0xE0, 0x5D	"apps" pressed				
0xE0, 0x9C	(keypad) enter released	0xE0, 0x9D	right control released				
		0xE0, 0xB5	(keypad) / released				
0xE0, 0xB8	right alt (or altGr) released						
						0xE0, 0xC7	home released
0xE0, 0xC8	cursor up released	0xE0, 0xC9	page up released			0xE0, 0xCB	cursor left released
		0xE0, 0xCD	cursor right released			0xE0, 0xCF	end released
0xE0, 0xD0	cursor down released	0xE0, 0xD1	page down released	0xE0, 0xD2	insert released	0xE0, 0xD3	delete released
						0xE0, 0xDB	left GUI released
0xE0, 0xDC	right GUI released	0xE0, 0xDD	"apps" released				
						0xE0, 0x2A, 0xE0, 0x37	print screen pressed
						0xE0, 0xB7, 0xE0,	print screen

						0xAA	released
		0xE1, 0x1D, 0x45, 0xE1, 0x9D, 0xC5	pause pressed				

Note: There is no scan code for "pause key released" (it behaves as if it is released as soon as it's pressed)

Scan Code Set 2

The following table shows which "make" scan codes correspond to which keys when using scan code set 2 (for a "US QWERTY" keyboard only):

Scan code	Key	Scan code	Key	Scan code	Key	Scan code	Key
		0x01	F9 pressed			0x03	F5 pressed
0x04	F3 pressed	0x05	F1 pressed	0x06	F2 pressed	0x07	F12 pressed
		0x09	F10 pressed	0x0A	F8 pressed	0x0B	F6 pressed
0x0C	F4 pressed	0x0D	tab pressed	0x0E	` (back tick) pressed		
		0x11	left alt pressed	0x12	left shift pressed		
0x14	left control pressed	0x15	Q pressed	0x16	1 pressed		
				0x1A	Z pressed	0x1B	S pressed
0x1C	A pressed	0x1D	W pressed	0x1E	2 pressed		
		0x21	C pressed	0x22	X pressed	0x23	D pressed
0x24	E pressed	0x25	4 pressed	0x26	3 pressed		
		0x29	space pressed	0x2A	V pressed	0x2B	F pressed
0x2C	T pressed	0x2D	R pressed	0x2E	5 pressed		
		0x31	N pressed	0x32	B pressed	0x33	H pressed
0x34	G pressed	0x35	Y pressed	0x36	6 pressed		
				0x3A	M pressed	0x3B	J pressed
0x3C	U pressed	0x3D	7 pressed	0x3E	8 pressed		
		0x41	, pressed	0x42	K pressed	0x43	I pressed
0x44	O pressed	0x45	0 (zero) pressed	0x46	9 pressed		
		0x49	. pressed	0x4A	/ pressed	0x4B	L pressed
0x4C	; pressed	0x4D	P pressed	0x4E	- pressed		
				0x52	' pressed		
0x54	[pressed	0x55	= pressed				
0x58	CapsLock pressed	0x59	right shift pressed	0x5A	enter pressed	0x5B	] pressed
		0x5D	\ pressed				
				0x66	backspace pressed		
		0x69	(keypad) 1 pressed			0x6B	(keypad) 4 pressed
0x6C	(keypad) 7 pressed						
0x70	(keypad) 0 pressed	0x71	(keypad) . pressed	0x72	(keypad) 2 pressed	0x73	(keypad) 5 pressed
0x74	(keypad) 6 pressed	0x75	(keypad) 8 pressed	0x76	escape pressed	0x77	NumberLock pressed
0x78	F11 pressed	0x79	(keypad) + pressed	0x7A	(keypad) 3 pressed	0x7B	(keypad) - pressed
0x7C	(keypad) * pressed	0x7D	(keypad) 9 pressed	0x7E	ScrollLock pressed		
						0x83	F7 pressed
0xE0, 0x10	(multimedia) WWW search pressed	0xE0, 0x11	right alt pressed				
0xE0, 0x14	right control pressed	0xE0, 0x15	(multimedia) previous track pressed				
0xE0, 0x18	(multimedia) WWW favourites pressed						
						0xE0, 0x1F	left GUI pressed
0xE0, 0x20	(multimedia) WWW refresh pressed	0xE0, 0x21	(multimedia) volume down pressed			0xE0, 0x23	(multimedia) mute pressed

						0xE0, 0x27	right GUI pressed
0xE0, 0x28	(multimedia) WWW stop pressed					0xE0, 0x2B	(multimedia) calculator pressed
						0xE0, 0x2F	apps pressed
0xE0, 0x30	(multimedia) WWW forward pressed			0xE0, 0x32	(multimedia) volume up pressed		
0xE0, 0x34	(multimedia) play/pause pressed					0xE0, 0x37	(ACPI) power pressed
0xE0, 0x38	(multimedia) WWW back pressed			0xE0, 0x3A	(multimedia) WWW home pressed	0xE0, 0x3B	(multimedia) stop pressed
						0xE0, 0x3F	(ACPI) sleep pressed
0xE0, 0x40	(multimedia) my computer pressed						
0xE0, 0x48	(multimedia) email pressed			0xE0, 0x4A	(keypad) / pressed		
		0xE0, 0x4D	(multimedia) next track pressed				
0xE0, 0x50	(multimedia) media select pressed						
				0xE0, 0x5A	(keypad) enter pressed		
				0xE0, 0x5E	(ACPI) wake pressed		
		0xE0, 0x69	end pressed			0xE0, 0x6B	cursor left pressed
0xE0, 0x6C	home pressed						
0xE0, 0x70	insert pressed	0xE0, 0x71	delete pressed	0xE0, 0x72	cursor down pressed		
0xE0, 0x74	cursor right pressed	0xE0, 0x75	cursor up pressed				
				0xE0, 0x7A	page down pressed		
		0xE0, 0x7D	page up pressed				
		0xF0, 0x01	F9 released			0xF0, 0x03	F5 released
0xF0, 0x04	F3 released	0xF0, 0x05	F1 released	0xF0, 0x06	F2 released	0xF0, 0x07	F12 released
		0xF0, 0x09	F10 released	0xF0, 0x0A	F8 released	0xF0, 0x0B	F6 released
0xF0, 0x0C	F4 released	0xF0, 0x0D	tab released	0xF0, 0x0E	` (back tick) released		
		0xF0, 0x11	left alt released	0xF0, 0x12	left shift released		
0xF0, 0x14	left control released	0xF0, 0x15	Q released	0xF0, 0x16	1 released		
				0xF0, 0x1A	Z released	0xF0, 0x1B	S released
0xF0, 0x1C	A released	0xF0, 0x1D	W released	0xF0, 0x1E	2 released		
		0xF0, 0x21	C released	0xF0, 0x22	X released	0xF0, 0x23	D released
0xF0, 0x24	E released	0xF0, 0x25	4 released	0xF0, 0x26	3 released		
		0xF0, 0x29	space released	0xF0, 0x2A	V released	0xF0, 0x2B	F released
0xF0, 0x2C	T released	0xF0, 0x2D	R released	0xF0, 0x2E	5 released		
		0xF0, 0x31	N released	0xF0, 0x32	B released	0xF0, 0x33	H released
0xF0, 0x34	G released	0xF0, 0x35	Y released	0xF0, 0x36	6 released		
				0xF0, 0x3A	M released	0xF0, 0x3B	J released
		0xF0,					

0xF0, 0x3C	U released	0x3D	7 released	0xF0, 0x3E	8 released		
		0xF0, 0x41	, released	0xF0, 0x42	K released	0xF0, 0x43	I released
0xF0, 0x44	O released	0xF0, 0x45	0 (zero) released	0xF0, 0x46	9 released		
		0xF0, 0x49	. released	0xF0, 0x4A	/ released	0xF0, 0x4B	L released
0xF0, 0x4C	; released	0xF0, 0x4D	P released	0xF0, 0x4E	- released		
				0xF0, 0x52	' released		
0xF0, 0x54	[released	0xF0, 0x55	= released				
0xF0, 0x58	CapsLock released	0xF0, 0x59	right shift released	0xF0, 0x5A	enter released	0xF0, 0x5B	] released
		0xF0, 0x5D	\ released				
				0xF0, 0x66	backspace released		
		0xF0, 0x69	(keypad) 1 released			0xF0, 0x6B	(keypad) 4 released
0xF0, 0x6C	(keypad) 7 released						
0xF0, 0x70	(keypad) 0 released	0xF0, 0x71	(keypad) . released	0xF0, 0x72	(keypad) 2 released	0xF0, 0x73	(keypad) 5 released
0xF0, 0x74	(keypad) 6 released	0xF0, 0x75	(keypad) 8 released	0xF0, 0x76	escape released	0xF0, 0x77	NumberLock released
0xF0, 0x78	F11 released	0xF0, 0x79	(keypad) + released	0xF0, 0x7A	(keypad) 3 released	0xF0, 0x7B	(keypad) - released
0xF0, 0x7C	(keypad) * released	0xF0, 0x7D	(keypad) 9 released	0xF0, 0x7E	ScrollLock released		
						0xF0, 0x83	F7 released
0xE0, 0x12, 0xE0, 0x7C	print screen pressed						
0xE0, 0xF0, 0x10	(multimedia) WWW search released	0xE0, 0xF0, 0x11	right alt released				
0xE0, 0xF0, 0x14	right control released	0xE0, 0xF0, 0x15	(multimedia) previous track released				
0xE0, 0xF0, 0x18	(multimedia) WWW favourites released						
						0xE0, 0xF0, 0x1F	left GUI released
0xE0, 0xF0, 0x20	(multimedia) WWW refresh released	0xE0, 0xF0, 0x21	(multimedia) volume down released			0xE0, 0xF0, 0x23	(multimedia) mute released
						0xE0, 0xF0, 0x27	right GUI released
0xE0, 0xF0, 0x28	(multimedia) WWW stop released					0xE0, 0xF0, 0x2B	(multimedia) calculator released
						0xE0, 0xF0, 0x2F	apps released
0xE0, 0xF0, 0x30	(multimedia) WWW forward released			0xE0, 0xF0, 0x32	(multimedia) volume up released		
0xE0, 0xF0, 0x34	(multimedia) play/pause released					0xE0, 0xF0, 0x37	(ACPI) power released
0xE0, 0xF0, 0x38	(multimedia) WWW back released			0xE0, 0xF0, 0x3A	(multimedia) WWW home released	0xE0, 0xF0, 0x3B	(multimedia) stop released
						0xE0, 0xF0, 0x3F	(ACPI) sleep released
0xE0, 0xF0, 0x40	(multimedia) my computer released						
0xE0, 0xF0, 0x48	(multimedia) email released			0xE0, 0xF0, 0x4A	(keypad) / released		
		0xE0, 0xF0, 0x4D	(multimedia) next track released				
0xE0, 0xF0,	(multimedia) media						

0x50	select released						
				0xE0, 0xF0, 0x5A	(keypad) enter released		
				0xE0, 0xF0, 0x5E	(ACPI) wake released		
		0xE0, 0xF0, 0x69	end released			0xE0, 0xF0, 0x6B	cursor left released
0xE0, 0xF0, 0x6C	home released						
0xE0, 0xF0, 0x70	insert released	0xE0, 0xF0, 0x71	delete released	0xE0, 0xF0, 0x72	cursor down released		
0xE0, 0xF0, 0x74	cursor right released	0xE0, 0xF0, 0x75	cursor up released				
				0xE0, 0xF0, 0x7A	page down released		
		0xE0, 0xF0, 0x7D	page up released				
				0xE0, 0xF0, 0x7C, 0xE0, 0xF0, 0x12	print screen released		
						0xE1, 0x14, 0x77, 0xE1, 0xF0, 0x14, 0xF0, 0x77	pause pressed

Note: There is no scan code for "pause key released" (it behaves as if it is released as soon as it's pressed)

Scan Code Set 3

The following table shows which scan codes correspond to which keys when using scan code set 3 (for a "US QWERTY" keyboard only):

TODO

See Also

- PS/2
- "8042" PS/2 Controller
- PL050 PS/2 Controller (ARM)
- PS/2 Mouse

Forum Threads

- Keyboard input
- Up or down press?
- Change typerate
- Converting the scancodes
- Discussion about keyboard input in a GUI
- Scroll-lock LED
- Keyboard LEDs (asm source)
- Keyboard LEDs (C source)

External Links

- www.Computer-Engineering.org (http://www.computer-engineering.org)
- KMT dk's ps2 keyboard and controller referance (http://www.webmasteren.eu/viden/os/PS2.pdf)
- Keyboard scancodes (http://www.win.tue.nl/~aeb/linux/kbd/scancodes.html) - A complete reference on all scancodes you might encounter.

Implementations

- Linux (http://lxr.linux.no/#linux+v3.5.4/drivers/input/keyboard/atkbd.c) (C,GPL)

Retrieved from "http://wiki.osdev.org/index.php?title=PS/2_Keyboard&oldid=16244"
Categories: Human Interface Device | Common Devices

- This page was last modified on 23 April 2014, at 23:03.
- This page has been accessed 164,176 times.