

PS2KEYBOARD DRIVER MODULE

[Suitable for 101 102 and 104 Keyboard]

Instruction:

Feature:

Converts the PS2 protocol to TWI/IIC,UART,parallel interface

Increase the key state detection. Wide range of applications.

Application : Robot ,Electronic products,and so on

1. Brief introduction

PS2 keyboard driver module, use a small microcontroller reads the PC PS2 keyboard key-press information, The values of the key will output by serial or parallel or TWI/IIC(Through the short circuit cap option), for the use of other equipment. In a single chip microcomputer project system, main control MCU does not need to write the keyboard driver, read the key information directly, very convenient.

2. Working principle

PS2 keyboard driver module uses a piece of programed code of MCU as the controller, to read the state of PS2 keyboard keys. When the keyboard is pressed, the LED lights flashing on the PS2 keyboard driver module, at the same time through serial communication and parallel I/o port or TWI/IIC(Through the short circuit cap option),Keys will be output.

For example, Press "A" key,serial port output data 0x1c, parallel I/o port output 0x1c by D7-D0. The output scan code can refer to the back of the content.

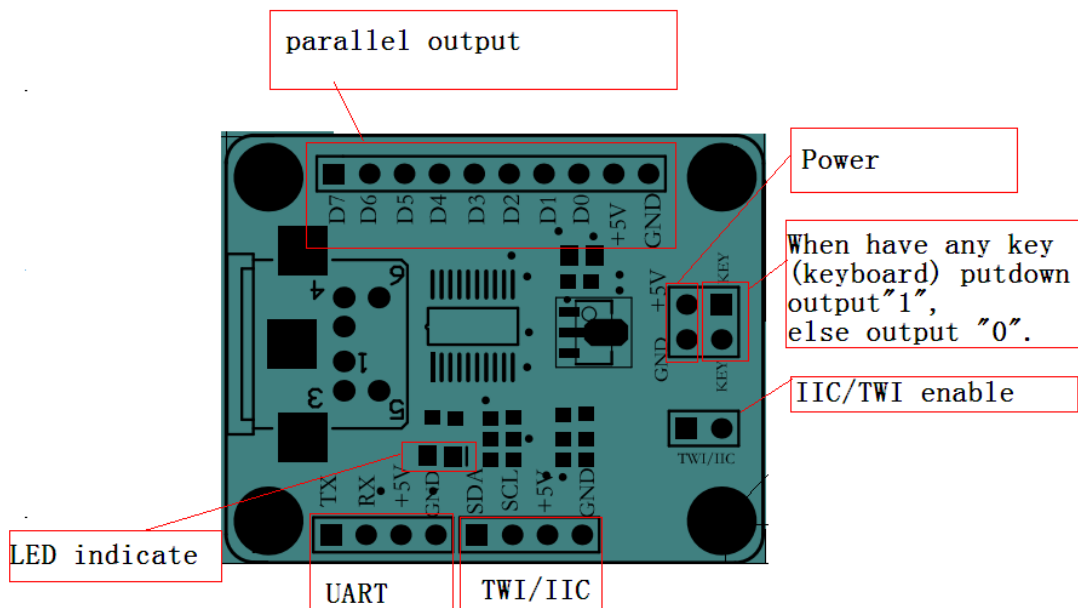
Pop-up output. In addition, two sets of buttons, TTL output when there is no buttons to press the "0", and have a button to press the output "1", after the button pop-up restore "0" (each key on keyboard effectively. In addition to pause/break key.

Serial, parallel, TWI/IIC can work at the same time. If there is no connection from the machine on the IIC/TWI bus, please remove short-circuit cap module above. Otherwise, the module will not be able to work properly.

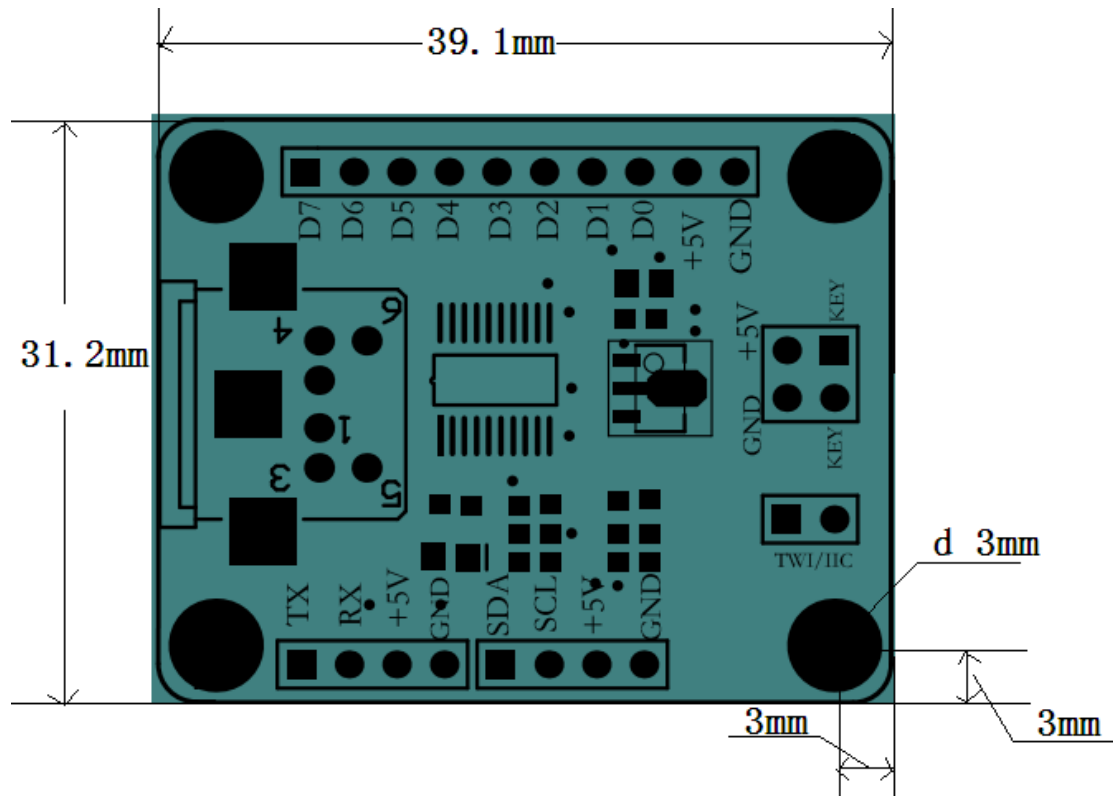
matters need attention:

1. TWI/IIC address is 4. **Modules for the host.**
2. Do not use the TWI/IIC should be "IIC/TWI enable" **Short circuit cap to remove.** Otherwise the module will not be able to work properly
- 3.*UART Baud rate: 19200

Function:

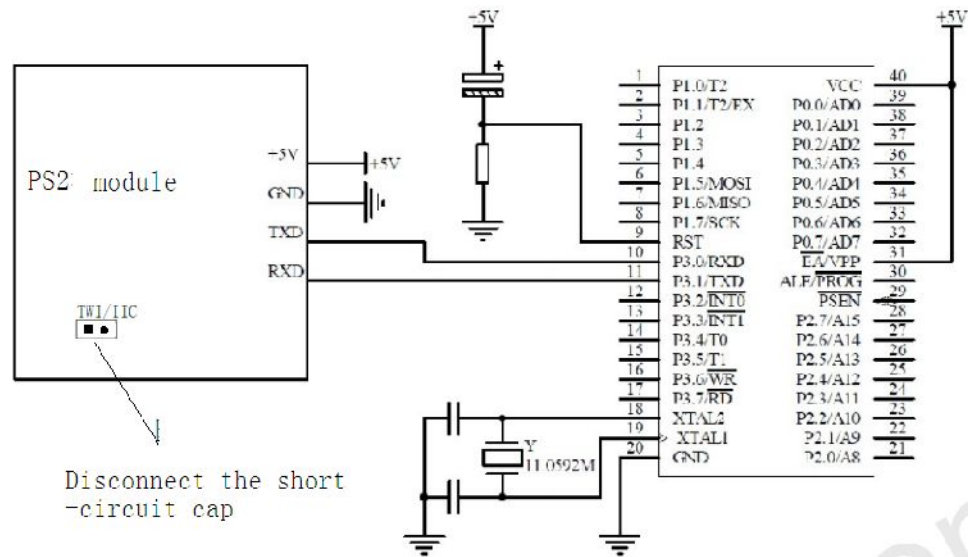


Note: All Power Linked together

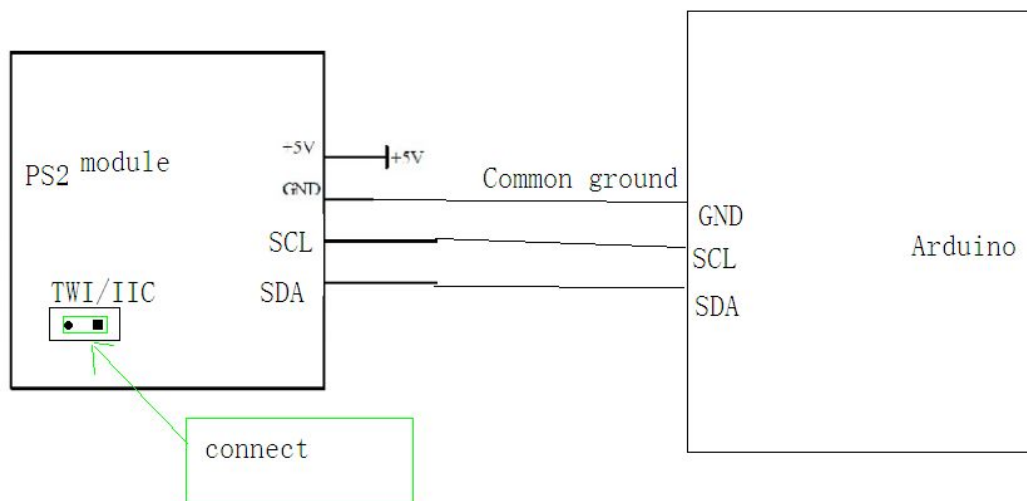


V Application specification

Using C51 serial mode circuit:



Using Arduino



Arduino Test code:

```
#include <Wire.h>
```

```
void setup()
```

```
{
```

```
  Wire.begin(4);           // Set slave model ,address is 4
```

```
  Wire.onReceive(receiveEvent);
```

```
  Serial.begin(9600);     // Initialize the serial port of 9600 baud rate
```

```
}
```

```
void loop()
```

```
{
```

```
  delay(100);
```

```
}
```

```
void receiveEvent(int howMany)
```

```
{
```

```
  unsigned char x = Wire.read(); // Read the data as an integer
```

```
  Serial.println(x); // Read data serial print
```

```
}
```

The output scan code table

KEY	UART Output	Parallel/TWI/IIC		KEY	UART Output	Parallel/TWI/IIC		KEY	UART Output	Parallel/TWI/IIC	
		Hexadecimal	Decimal			Hexadecimal	Decimal			Hexadecimal	Decimal
A	1C	1C	28	9	46	46	70	[54	54	84
B	32	32	50	`	0E	0E	14	INSERT	E0 70	A3	163
C	21	21	33	-	4E	4E	78	HOME	E0 6C	A4	164
D	23	23	35	=	55	55	85	PG UP	E0 7D	A5	165
E	24	24	36	\	5D	5D	93	DELETE	E0 71	A6	166
F	2B	2B	43	BKSP	66	66	102	END	E0 69	A7	167
G	34	34	52	SPACE	29	29	41	PG DN	E0 7A	A8	168
H	33	33	51	TAB	0D	0D	13	U ARROW	E0 75	A9	169
I	43	43	67	CAPS	58	58	88	L ARROW	E0 6B	AA	170
J	3B	3B	59	L SHFT	12	12	18	D ARROW	E0 72	AB	171
K	42	42	66	L CTRL	14	14	20	R ARROW	E0 74	AC	172
L	4B	4B	75	L GUI	E0 1F	A0	160	NUM	77	77	119
M	3A	3A	58	L ALT	11	11	17	KP /	E0 4A	AD	173
N	31	31	49	R SHFT	59	59	89	KP *	7C	7C	124
O	44	44	68	R CTRL	E0 14	A1	161	KP -	7B	7B	123
P	4D	4D	77	R GUI	E0 27	27	39	KP +	79	79	121
Q	15	15	21	R ALT	E0 11	A2	162	KP EN	E0 5A	AE	174
R	2D	2D	45	APPS	E0 2F	2F	47	KP	71	71	113
S	1B	1B	27	ENTER	5A	5A	90	KP 0	70	70	112
T	2C	2C	44	ESC	76	76	118	KP 1	69	69	105
U	3C	3C	60	F1	5	5	5	KP 2	72	72	114
V	2A	2A	42	F2	6	6	6	KP 3	7A	7A	122
W	1D	1D	29	F3	4	4	4	KP 4	6B	6B	107
X	22	22	34	F4	0C	0C	12	KP 5	73	73	115
Y	35	35	53	F5	3	3	3	KP 6	74	74	116
Z	1A	1A	26	F6	0B	0B	11	KP 7	6C	6C	108
0	45	45	69	F7	83	83	131	KP 8	75	75	117
1	16	16	22	F8	0A	0A	10	KP 9	7D	7D	125
2	1E	1E	30	F9	1	1	1]	58	58	88
3	26	26	38	F10	9	9	9	;	4C	4C	76
4	25	25	37	F11	78	78	120	'	52	52	82
5	2E	2E	46	F12	7	7	7	,	41	41	65
6	36	36	54	PRNT SCRN	E0 12 E0 7C	C4	196	.	49	49	73
7	3D	3D	61	SCROLL	7E	7E	126	/	4A	4A	74
8	3E	3E	62	PAUSE	E1 14 77 E1 F0 14 F0 77	C5	197				

The output scan code table

ACPI scan code:

KEY	UART Output	Parallel/TWI/IIC	
		Hexadecimal	Decimal
Power	E0, 37	AF	175
Sleep	E0, 3F	B0	176
Wake	E0, 5E	B1	177

Windows Media scan code:

KEY	UART Output	Parallel/TWI/IIC	
		Hexadecimal	Decimal
Next Track	E0, 4D	B2	178
Previous Track	E0, 15	B3	179
Stop	E0, 3B	B4	180
Play/Pause	E0, 34	B5	181
Mute	E0, 23	B6	182
Volume Up	E0, 32	B7	183
Volume Down	E0, 21	B8	184
Media Select	E0, 50	B9	185
E-Mail	E0, 48	BA	186
Calculator	E0, 2B	BB	187
My Computer	E0, 40	BC	188
WWW Search	E0, 10	BD	189
WWW Home	E0, 3A	BE	190
WWW Back	E0, 38	BF	191
WWW Forward	E0, 30	C0	192
WWW Stop	E0, 28	C1	193
WWW Refresh	E0, 20	C2	194
WWW Favorites	E0, 18	C3	195

Note: The code table is only suitable for 101 102 and 104 Keyboard