

# P6 Let's make a blog

Development diary of trying to realize PC-6001 with FPGA

The analysis articles, production articles, and remodeling articles in this blog are by amateurs. Since the article may be incorrect, it is the responsibility of the user to use the content (we are not responsible).

<< Pocky for P6 | TOP | About the 66SR keyboard (2) >>

November 15, 2014

## About 66SR keyboard (1)

It's about the 66SR keyboard, which has been a hot topic recently.

There is information in MORIYA's place, but I re-examined it and summarized it.

### 1. 1. About the method of exchanging data Since the

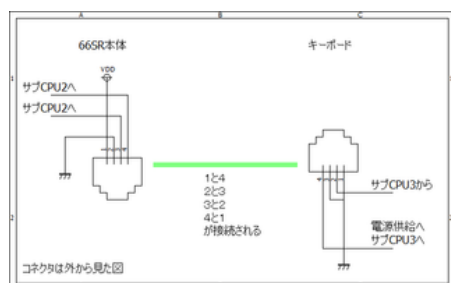
66SR and the keyboard are separated, the keyboard information is transmitted by infrared communication or wired communication. Data is unilaterally sent from the keyboard to 66SR.

Only infrared communication or wired communication is used, but the selection method is as follows.

- If the 66SR main unit and keyboard are connected with the attached cable, wired communication
- Other than that, infrared communication

The electrical selection method in the actual circuit is as follows.

**(Corrected the circuit diagram) When the** 66SR main unit and the keyboard are connected with a cable, the keyboard side



~~2 pins~~Pin 4 becomes "H", the result is detected by the sub CPU3, and the transmission circuit is switched from infrared communication to wired communication.

Also, on the 66SR side~~3 pins~~Pin 4 becomes "L", the result is detected by sub CPU2, and the receiving circuit is switched to wired communication.

The actual transmitted data is~~it is sent from the keyboard to 66SR using 4 pins~~It is transmitted from pin 2 of the keyboard to pin 3 of the 66SR main unit.

### 2. 2. About data format

Infrared communication and wired communication are transmitted using the same logical format.

However, the actual physical format is different (see below).

One data transmission is 11-bit data of 3-bit control data (CT [2: 0]) + 8-bit key data (DT [7: 0]).

The logical format is as follows. (Be careful of LSB postponement) When you press the key once, the code (control data + key data) of the pressed key

<< October 2020 >>

Day	Moon	fire	water	wood	Money	soil
				1	2	3
Four	Five	6	7	8	9	Ten
11	12	13	14	15	16	17
18	19	20	twenty one	twenty two	twenty three	twenty four
twenty five	26	27	28	29	30	31

### Search in Blog

### View the latest articles

- (01/01) [Contents](#)
- (10/28) [ROM file selection program for NANOCART \(ver1.1\)](#)
- (10/28) [NANOCART release \(ver1.21\)](#)
- (10/27) [Problems when using mega ROM for anything with NANOCART](#)
- (10/25) [Modification using SD6031WIF / NANOCART with mk2 / mk2SR](#)
- (10/24) [NANOCART / Warrior mk2 similar non- mk2 mode](#)
- (10/24) [ROM file selection program for NANOCART \(ver1.0\)](#)
- (10/24) [NANOCART Release \( ver1.20 \)](#)
- (10/11) [SD6031 related](#)
- (10/11) [Image file selection program for FlashAir \(ver1.1\)](#)
- (10/11) [Image file selection program for SD6031 etc. \(ver3.0\)](#)
- (10/11 ) ) [SD6031Air firmware release \(ver1.20\)](#)
- (10/11) [SD6031 firmware release \(ver1.20\)](#)
- (08/14) [SR support for ROMization of door door mk2 \(3\)](#)
- (08/13) [Door door mk2 ROMized SR compatible \(2\)](#)
- (08/10) [SD6031 Firmware release \(ver1.13\)](#)
- (08/09) [Door door mk2 ROMized SR compatible \(06/29\)](#)
- [FM-TOWNS CD drive repair \( 06/18\)](#)
- [Power supply repair of the first machine \(05/19\)](#)
- [Addition of RGB converter \(15\) \(DE0 version ver32 release\)](#)

### View latest comments

- [SD6031 Firmware release \(ver1.20\)](#)
  - ⇒ MORIYA (12/07)
  - ⇒ Esubi (12/07)
  - ⇒ Esubi (12/07)
  - ⇒ MORIYA (12/06)
- [P6 floppy related summary \(4\)](#)
  - ⇒ Esubi (12/06)

is sent. If you hold it down, the keycode will be sent continuously. Auto repeat is realized on the keyboard side. When you release the key, the key code will not be sent. It is a list of key codes. The first 3 digits are the 3-bit control code (CT [2: 0]), and the last 2 digits are the key data (DT [7: 0] / HEX notation). After that, the key data is converted to characters. For example, in the case of 000: 30 (0), the control code is 0b000, the key data is 0x30, and the ASCII code is "0". "None" is the data sent when no other key is pressed. "SHIFT" is the data sent when you hold down the SHIFT key and press the key. The same applies to "CTRL" and "GRAPH". "TV" is the data that is sent when you press the "PC / TV" button at the top of the keyboard and then press the key. There are blanks because no other keys will send data.

ヘッダ	CT0	CT1	CT2	DT0	DT1	DT2	DT3	DT4	DT5	DT6	DT7	PTY	Q00	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08	Q09	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

CT[2:0] = 制御データ      DT[7:0] = キーデータ      アンダーラインの信号は、元の信号の反転を示す

PTY = CT[2:0]+DT[7:0] の奇数/リディ      CT[2:0]とDT[7:0]の"1" の数が奇数なら、奇数/リディが"1"になる

⇒ bookworm (12/06)

Circuit diagram of

pc6601SR ⇒ Esubi (12/06)

⇒ leaded solder (12/06)

Video playback release (1)

⇒ yo (11/24)

⇒ Esubi (11 ) / 24)

⇒ yo (11/24)

FM-TOWNS CD drive repair

⇒ Esubi (10/03)

⇒ Roki-chan (10/03)

⇒ WINDY (09/09)

⇒ Esubi (09/09)

⇒ WINDY (09/09)

## Archive by category

FPGA: Release (43)

No category (1)

P6 analysis: Circuit in general (24)

P6 analysis: Screen circumference (41)

P6 analysis: Sub CPU and 8255 (12)

P6 analysis: Audio circumference (4)

P6 analysis: Joystick Port (5)

P6 analysis: around CPU (12)

P6 analysis: around FDC (18)

P6 analysis: voice synthesis (12)

P6 analysis: RS232C (1)

P6 analysis: Kanji (3)

P6 analysis: Other (5)

FPGA

FPGA : General circuit (7) FPGA: CPU (9)

FPGA: Screen (10)

FPGA: Sub CPU and 8255 (9)

FPGA: Audio (2)

FPGA: Joystick port (4)

FPGA Conversion: Around FDC (6)

FPGA conversion: Voice synthesis (0)

FPGA: RS232C (3)

FPGA: SDRAM (5)

FPGA: SD card (10)

FPGA: Clock system (1)

FPGA: Other (5)

Tool: DE0 / Quartus (8)

Tool: Other (2 ) )

SW: CP / M (2)

SW: Tool (8)

SW: Porting / Modification / Analysis / Debugging etc. (140)

SW: Others (3)

HW: AVR (0)

HW: RS232C board (11)

HW: Video Playback (18)

HW: SD6031 (45)

HW: Others (38)

P6 Actual machine: Manufacture / Repair / Modification (29)

Others (15)

## monthly archive

10 May 2020 (12)

08 January 2020 (4)

06 June 2020 (2)

May 2020 (9)

キー	なし	+SHIFT	+CTRL	+GRAPH	TV
O	000:30(0)	000:A6( )	000:00( )	001:0F(万)	011:19( )
1	000:31(1)	000:21(!)	000:00( )	001:07(日)	011:10( )
2	000:32(2)	000:22(~)	000:00( )	001:01(月)	011:11( )
3	000:33(3)	000:23(#)	000:00( )	001:02(火)	011:12( )
4	000:34(4)	000:24(\$)	000:00( )	001:03(水)	011:13( )
5	000:35(5)	000:25(%)	000:00( )	001:04(木)	011:14( )
6	000:36(6)	000:26(&)	000:00( )	001:05(金)	011:15( )
7	000:37(7)	000:27(^)	000:00( )	001:06(土)	011:16( )
8	000:38(8)	000:28( )	000:00( )	001:00(百)	011:17( )
9	000:39(9)	000:29( )	000:00( )	001:0E(千)	011:18( )
A	000:61(a)	000:41(A)	000:01( )	001:00( )	
B	000:62(b)	000:42(B)	000:02( )	001:1B(↓)	
C	000:63(c)	000:43(C)	000:03( )	001:1A(↵)	
D	000:64(d)	000:44(D)	000:04( )	001:14(ト)	
E	000:65(e)	000:45(E)	000:05( )	001:18(↑)	
F	000:66(f)	000:46(F)	000:06( )	001:15(十)	
G	000:67(g)	000:47(G)	000:07( )	001:13(↑)	
H	000:68(h)	000:48(H)	000:08( )	001:0A(時)	
I	000:69(i)	000:49(I)	000:09( )	001:16(   )	
J	000:6A(j)	000:4A(J)	000:0A( )	001:00( )	
K	000:6B(k)	000:4B(K)	000:0B( )	001:00( )	
L	000:6C(l)	000:4C(L)	000:0C( )	001:1E(中)	
M	000:6D(m)	000:4D(M)	000:0D( )	001:0B(分)	
N	000:6E(n)	000:4E(N)	000:0E( )	001:00( )	
O	000:6F(o)	000:4F(O)	000:0F( )	001:00( )	
P	000:70(p)	000:50(P)	000:10( )	001:10(π)	
Q	000:71(q)	000:51(Q)	000:11( )	001:00( )	
R	000:72(r)	000:52(R)	000:12( )	001:12(丁)	
S	000:73(s)	000:53(S)	000:13( )	001:0C(秒)	
T	000:74(t)	000:54(T)	000:14( )	001:19(ㄣ)	
U	000:75(u)	000:55(U)	000:15( )	001:00( )	
V	000:76(v)	000:56(V)	000:16( )	001:11(上)	
W	000:77(w)	000:57(W)	000:17( )	001:00( )	
X	000:78(x)	000:58(X)	000:18( )	001:1C(×)	
Y	000:79(y)	000:59(Y)	000:19( )	001:08(年)	
Z	000:7A(z)	000:5A(Z)	000:1A( )	001:00( )	
空白	000:20( )	000:20( )	000:00( )	001:00( )	
,	000:2C(,)	000:3C(<)	000:00( )	001:1F(小)	
-	000:2D(-)	000:3D(=)	000:00( )	001:17(一)	011:1A( )
.	000:2E(.)	000:3E(>)	000:00( )	001:1D(大)	
/	000:2F(/)	000:3F(?)	000:00( )	001:80( )	
@	000:40(@)	000:40(@)	000:00( )	001:00( )	
[	000:5B([)	000:A2( )	000:1B( )	001:84( )	
]	000:5D(])	000:A3( )	000:1D( )	001:85( )	
^	000:5E(^)	000:00( )	000:1E( )	001:00( )	011:1B( )
¥	000:0B( )	000:5F( )	000:00( )	001:83( )	
¥	000:5C(¥)	000:00( )	000:1C( )	001:09(円)	
:	000:3A(:)	000:2A(*)	000:00( )	001:81( )	
:	000:3B(:)	000:2B(+)	000:00( )	001:82( )	
TAB	000:09( )	000:09( )	000:00( )	001:00( )	
HOME	000:0C( )	000:0B( )	000:0C( )	000:0C( )	011:0A( )
RETURN	000:0D( )	000:0D( )	000:00( )	001:00( )	
INS	000:12( )	000:12( )	000:12( )	000:12( )	
ESC	000:1B( )	000:1B( )	000:00( )	001:00( )	
→	000:1C( )	000:1C( )	000:00( )	001:00( )	
←	000:1D( )	000:1D( )	000:00( )	010:20( )	
↑	000:1E( )	000:1E( )	000:00( )	001:00( )	
↓	000:1F( )	000:1F( )	000:00( )	010:08( )	
DEL	000:7F( )	000:7F( )	000:7F( )	000:7F( )	011:1C( )
PAGE	000:FE( )	001:FC( )	000:FE( )	000:FE( )	011:04( )
F1	001:F0( )	001:F5( )	001:00( )	001:F0( )	011:00( )
F2	001:F1( )	001:F6( )	001:00( )	001:F1( )	011:01( )
F3	001:F2( )	001:F7( )	001:00( )	001:F2( )	011:02( )
F4	001:F3( )	001:F8( )	001:00( )	001:F3( )	011:03( )
F5	001:F4( )	001:F9( )	001:00( )	001:F4( )	011:0F( )
STOP	001:FA( )	001:FA( )	010:02( )	001:00( )	
CAPS	001:FB( )	001:FB( )	001:FB( )	001:FB( )	
MODE	001:FD( )	001:FD( )	001:FD( )	001:FD( )	
かな	001:FE( )	001:FE( )	001:FE( )	001:FE( )	
電源	100:08( )	100:08( )	100:08( )	100:08( )	

[04 April 2020](#) (2)  
[03 May 2020](#) (1)  
[02 January 2020](#) (2)  
[2020 January](#)  
[2019](#) (2) [December 2019](#) (1) [July 2019](#) (5)  
[April](#)  
[2019](#) (2) [February 2019](#) (2)  
[December 2018](#) (10)  
[November 2018](#) (5)  
[2018 May 10](#) (3)  
[08 June 2018](#) (2)  
[07 July 2018](#) (2)  
[06 June 2018](#) (30)  
[04 May 2018](#) (7)  
[03 May 2018](#) (6)

There is no "SHIFT", "CTRL", or "GRAPH" in the key, because this key does not send data by itself.

Even if you press the "CAPS" and "Kana" keys, the key code sent does not change. It seems that the sub CPU on the main unit is performing the conversion of hiragana / katakana.

In addition to the key code, the data for the game key will be sent.

Game keys are 7 keys: STOP, SPACE, SHIFT, cursor up / down / left / right.

Unlike other keys, these keys tell you whether they are pressed or released. Other keys only know that they have been pressed.

The transmitted data for the game key is as follows.

Control code (CT [2: 0]) = 0b010

Key data (DT [7: 0]) = Information of the

key being pressed Each bit of the key data is as follows.

DT [7]: SPACE

DT [6]: Fixed to "0"

DT [5]: ←

DT [4]: →

DT [3]: ↓

DT [2]: ↑

DT [1]: STOP

DT [0]: SHIFT

Outputs to the pressed key "1" and the non-pressed key "0", and when multiple game keys are pressed, the corresponding bit is "1". Will be. If all game keys are pressed, it will be 0xBF.

The internal processing seems to be monitoring the monitoring for the game key by polling. It seems that we are checking the state where the game key is pressed / not pressed at regular intervals.

At that time, if it is different from the previous state, it will be sent as transmission data for the game key. Normally, it is sent when the game key is pressed or released.

The transmission for the game key is done independently of the normal key code above. Therefore, when STOP, SPACE, or the cursor key is pressed, both the game key transmission and the key code transmission are sent.

For example, if you press the SPACE key when nothing is pressed, two data will be sent

: 0b010: 0x80 (send for game key)

0b000: 0x20 (send key code)

.

Also, it usually seems that the transmission for the game key comes first, but it can be reversed.

Continued next.

posted by Esubi at 14:06 | [Comment \(10\)](#) | [P6 Analysis: Sub CPU and 8255](#) |  

#### Comments on this article

はじめまして。にがと申します。普段はMSXを弄っている人ですが、赤外線ネタが好きすぎて、PC-6601SRのキーボード無しジャンクを入手し、MSXをキーボード代用品にしてみました。

その動画がコチラです。

[https://youtu.be/UO2k\\_8vTPTo](https://youtu.be/UO2k_8vTPTo)

MSX用のプログラムを書くにあたり、貴殿の解析情報を大いに活用させていただきました。ありがとうございました。ところで、解析情報のGRAPH+左/下のデータですが、特殊コード010:20, 010:08 はゲーム用の出力データを観測されたものと思います。自作の変換プログラムではGRAPH+上/右同様に001:00としました。

これを機にP6も少し弄ってみようかなと思っています。

Posted by にが at 2020年03月08日 13:56

報告ありがとうございます。

ご指摘の通り、GRAPH+←、↓は、001:00 を出力するのが自然ですね。そのうち修正しておきます。

Posted by えすび at 2020年03月08日 14:39

先日の動画でご報告させていただいたMSXによるPC-6601SR用赤外線キーボードですが、せっかくなので専用のカートリッジ基板を作ってみました。同人ハードとしてリリースしたいと考えていますが、問題ございますでしょうか。

取扱説明書を書いてみました。

<http://niga2.sytes.net/upfile/p6key.pdf>

Posted by にが at 2020年08月04日 18:07

作成も配布も特に問題ありません。

MSXが、P6のスレーブ（奴隷）になるのが、世の中のP6erに受けるかも(^^;)

キーマトリックスの関係から、カーソルキーの同時押しが検出できるので、MSXを使うのはベターな解ですね。

普通のキーボードだと、どうしてもカーソルキーの同時押しの検出不能問題が出てくるんですよ。

Posted by えすび at 2020年08月04日 20:08

ありがとうございます。近いうちに家電のケンちゃんさんを介してリリースしたいと思います。66SRユーザーの方がMSXを入手する機会になるかもしれませんし、お互いWinWinではないかと。

実はファミベキーボード用の送信機も作りました。ゲームキー同時入力含めて全キー押下できますが、MSXと違って任意のキーの状態を任意のタイミングで取得できないため、ややレスポンスは遅いです。需要がニッチすぎるのとガワ加工も面倒なので製品化の予定はありません。

<http://niga2.sytes.net/upfile/FB66SRKEY.jpg>

有線接続アダプタならアリかなと思ったのですが、66SRのコネクタに普通のモジュラー4pが刺さらないため断念しました。あれって専用品なのでしょうか。

Posted by にが at 2020年08月04日 21:25

4 P モジュラは汎用品です。

電話機の、受話器と本体をつないでいるコードに使われています。

電話機の回線側のモジュラジャックより、ひとまわり小さいサイズです。

ファミベのキーボードを使う案はいいですね。私も作ってみようかな...

Posted by えすび at 2020年08月04日 23:26

電話のモジュラーは6p、2芯、66SRは4p、4芯ですよ。

ところが4pのモジュラーが刺さらないのです。手持ちのモジュラープラグが変なのかも知れませんが、普通にどこかのパーツ屋さんから買ったものだったような気がします。

本体を改造してジャックを取り替えれば接続できますが、それだと同人ハードとして成り立たなくなり、作る意味がないかなと思っています。

Posted by にが at 2020年08月04日 23:40

以下で入手できそうです。

モジュラジャック

→アマゾンで「モジュラージャック 4P4C」で検索。

接続用コード

→アマゾンで「受話器コード」で検索。

コードに関しては、ホームセンターなどでも入手可能だと思います（私はコーナンで買いました）。

Posted by えすび at 2020年08月05日 20:16

改めて電話の受話器ケーブルを外して挿し込んでみたところ、なんと普通に挿さりました。最初に試したのはモジュラプラグをカシメる前の状態で、どうやら電極が飛び出しているために挿さらないみたいです。お騒がせしました。

有線で繋がりそうなので、試作してみることにします。赤外線バージョンは概ねできていたのでタイミングを少し弄るだけです。

えすびさんお作りになるのであればプログラム提供いたしますが、PIC（16F1822）の書き込み環境はお持ちでしょうか。

Posted by なが at 2020年08月08日 01:51

AVRの環境はありますが、PICは使っていません。

構造も簡単そうなので、自分で組もうかと思っています。

Posted by えすび at 2020年08月08日 15:51

コメントを書く

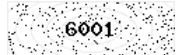
Name: [Required]

mail address:

homepage address:

Comment: [Required]

Authorization code: [Required]



\* Please enter the characters in the image in half-width characters.

Confirm

write

powered by Seesaa

Seesaa blog