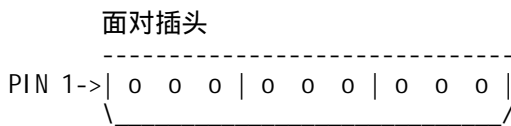


索尼 PLAYSTATION® 手柄原理分析

目录

- [Playstation 手柄针脚输出](#)
- [PS手柄信号](#)
- [PS手柄数据](#)
- [用74XX逻辑电路仿真PS手柄](#)
- [用微处理器仿真PS手柄](#)

PS手柄针脚输出



针脚 # 作用

1. DATA
2. COMMAND
3. N/C (9 Volts unused)
4. GND
5. VCC
6. ATT
7. CLOCK
8. N/C
9. ACK

DATA

信号流向从手柄到主机。

此信号是一个8 bit的串行数据，同步传送于时钟下降沿(输入输出信号在时钟信号由高到低时变化，所有信号的读取在时钟前沿到电平变化之前完成。)

COMMAND

信号流向从主机到手柄。

此信号和DATA相对，同样是一个8 bit的串行数据，同步传送于时钟下降沿。

VCC

电源电压从5V到3V原装的索尼手柄都可以工作。

主机主板上装有表面安装的750mA 保险丝，用于防止外设过载(750mA是包括左右手柄和记忆卡)。

ATT

ATT 用于提供手柄触发信号。

信号在通信期间处于低电平。又有人将此针脚叫做 Select, DTR 和 Command。

CLOCK

信号流向从主机到手柄。

用于保持数据同步。

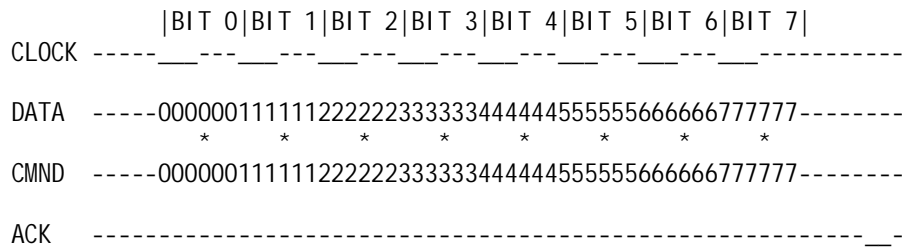
ACK

从手柄到主机的应答信号。

此信号在每个8 bits数据发送之后的最后一个时钟周期变低，并且ATT 一直保低电平。如果ACK 信号不变低约60微秒PS主机会试另一个外设。

PS手柄信号

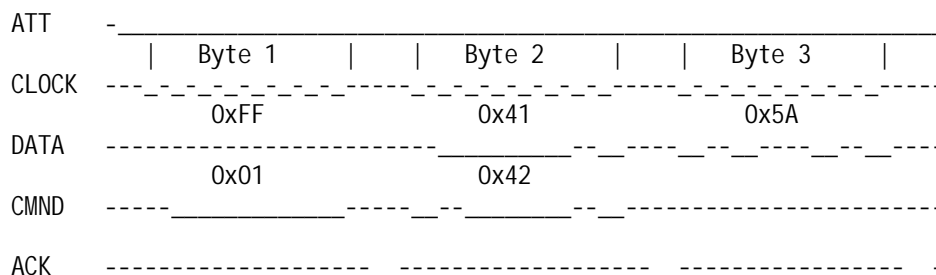
所有通讯都是8 bit串行数据最低有效位先行。在PS 手柄总线的所有时码在时钟下降沿都是同步的。传送一个字节的情况如下所示。



数据线的逻辑电平在时钟下降沿驱动下触发改变。数据的接收读取在时钟的前沿（在记号*处）到电平变化之前完成。在被选手柄接收每个COMMAND 信号之后，手柄需拉低ACK 电平在最后一个时钟。如果被选手柄没ACK 应答主机将假定没手柄接入。

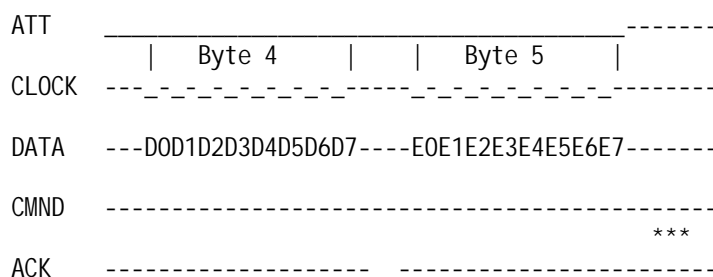
当PS主机想读一个手柄的数据时，将会拉低ATT 线电平并发出一个开始命令（0x01）。手柄将会回复它的ID（0x41=数字，0x23=NegCon，0x73=模拟红灯，0x53=模拟绿灯）。在手柄发送ID字节的同时主机将传送0x42 请求数据。随后命令线将空闲和手柄送出 0x5A 意思说：“数据来了”。

下面是一个数字手柄的时钟信号



在手柄执行初始化命令之后将发送它所有的数据字节（数字手柄只有两个字节）。在最后字节发送之后使ATT 高电平，手柄无需ACK应答。

数字手柄的数据传送如下所示(这里A0, A1, A2... B6, B7 是两个字节的数据比特)。



注意：没ACK.

PS手柄数据

下面五个表显示手柄的实际发送字节

标准数字手柄

BYTE	CMND	DATA	
01	0x01	idle	
02	0x42	0x41	
03	idle	0x5A	Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7
04	idle	data	SLCT STRT UP RGHT DOWN LEFT
05	idle	data	L2 R2 L1 R1 /\ 0 X _

所有按键按下有效。

NegCon

BYTE	CMND	DATA	
01	0x01	idle	
02	0x42	0x23	
03	idle	0x5A	Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7
04	idle	data	STRT UP RGHT DOWN LEFT
05	idle	data	R1 A B
06	idle	data	Steering 0x00 = Right 0xFF = Left
07	idle	data	I Button 0x00 = Out 0xFF = In
08	idle	data	II Button 0x00 = Out 0xFF = In
09	idle	data	L1 Button 0x00 = Out 0xFF = In

所有按键按下有效。

模拟手柄红灯模式

BYTE	CMND	DATA	
01	0x01	idle	
02	0x42	0x73	
03	idle	0x5A	Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7
04	idle	data	SLCT JOYR JOYL STRT UP RGHT DOWN LEFT
05	idle	data	L2 R2 L1 R1 /\ 0 X _
06	idle	data	Right Joy 0x00 = Left 0xFF = Right
07	idle	data	Right Joy 0x00 = Up 0xFF = Down
08	idle	data	Left Joy 0x00 = Left 0xFF = Right
09	idle	data	Left Joy 0x00 = Up 0xFF = Down

所有按键按下有效。

模拟手柄绿灯模式

BYTE	CMND	DATA	
01	0x01	idle	
02	0x42	0x53	
03	idle	0x5A	Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7
04	idle	data	STRT UP RGHT DOWN LEFT
05	idle	data	L2 L1 _ /\ R1 0 X R2
06	idle	data	Right Joy 0x00 = Left 0xFF = Right
07	idle	data	Right Joy 0x00 = Up 0xFF = Down
08	idle	data	Left Joy 0x00 = Left 0xFF = Right
09	idle	data	Left Joy 0x00 = Up 0xFF = Down

所有按键按下有效。

(credit to T. Fujita)

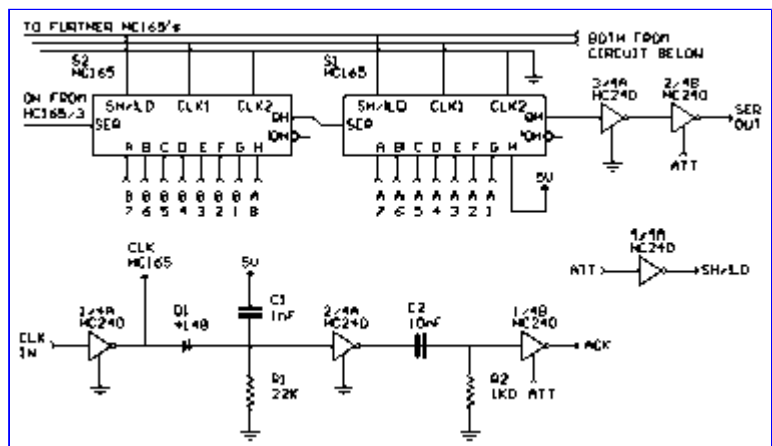
BYTE	CMND	DATA
01	0x01	i d l e
02	0x42	0x12
03	i d l e	0x5A B i t 0 B i t 1 B i t 2 B i t 3 B i t 4 B i t 5 B i t 6 B i t 7
04	i d l e	0xFF
05	i d l e	data L R
06	i d l e	data Del ta V e r t i c a l
07	i d l e	data Del ta H o r i z o n t a l

用74XX 逻辑电路仿真PS 手柄

这个电路能设置到仿真一个数字手柄，一个模拟手柄（两种模式）或一个NegCon。本电路使用了六块74XX IC仿真一个数字手柄，如仿真一个模拟手柄要多四块74XX IC'和四块A/D转换器，同样仿真一个NegCon手柄一样要多四块74XX IC'和四块A/D转换器。

电路如何工作

当ATT 电平被主机拉低，反相器4/4A 拉高74HC165的SH/!LD脚，串口输入脚读入数据(图中只画了五块IC中的二块)。反相器1/4A 为HC165 提供时钟信号，因此在PS 主机输出的CLK信号的每个下降沿HC165 移出数据的下一比特。数据经由ATT 信号控制的反相器2/4B门电路输出，使到只有被选的设备才能接入总线。



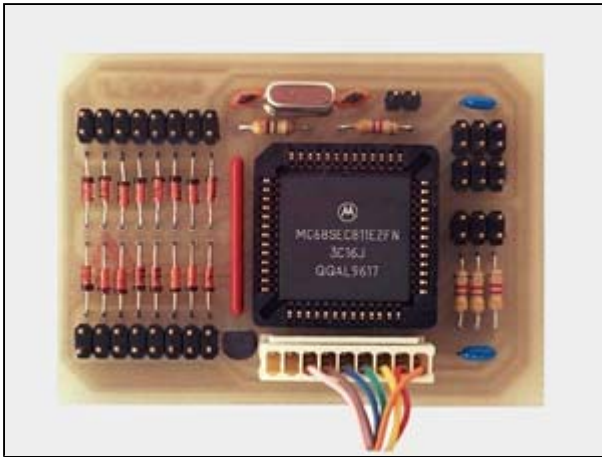
读进HC165 的数据的时码关系如以上图表所示。看回图表所示第一字节是0xFF，意思是从A1 到A8 全是高位。

下一个字节是手柄ID。数据手柄的ID是0x41 即B1-8 是HLLLLLHL 排列。再下一个字节是数据准备命令0x5A 即C1-8 为HLHLLHLH。最后两个字节是表示按键情况，当没按下为高电平，按下为低电平。

Ack 信号由反相器1/4A, 2/4A 和 1/4B产生。二极管D1 只允许当CLK低时反相器1/4A对C1 充电。当CLK 高位时间比R1/C1 设定的长, 反相器2/4A 将输出高电平。高电平经过C2到反相器1/4B 输出低脉冲, 由ATT 信号控制门电路输出(避免争夺总线)。

用微处理器仿真PS 手柄

微处理器使用Motorola 68HC11。运行代码使用汇编，可内置或使用任何52脚的EEPROM。



电路如何工作

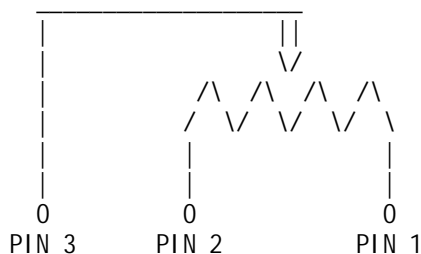
本电路使用一片Motorola MC68HC11完成所有工作。状态的读入使用16路N/O (normaly open) 开关输入，在电路的左边。四路模拟输入在右边。数据发送使用了芯片两个串行口中的一个。

四个跳线控制电路的工作情况。J1 跳线在电路板顶部控制HC11 工作在两种操作模式中的一种。当J1 打开时微处理器运行在单芯片模式，闭合时在特殊的bootstrap 模式。

J2, 3 和4 选择仿真何种PS手柄，如下表所示。

模式	J2	J3	J4
数字	0	0	0
模拟红灯	0	0	C
NegCon	0	C	0
模拟绿灯	0	C	C
保留	C	X	X
0 = 打开, C = 关闭, X = 不用			

四个模拟输入在电路的右上角设计外接电位器。电位器值并没规定一般用10K 到50K 较灵敏。三条线分别接中点，电源和地如下所示。



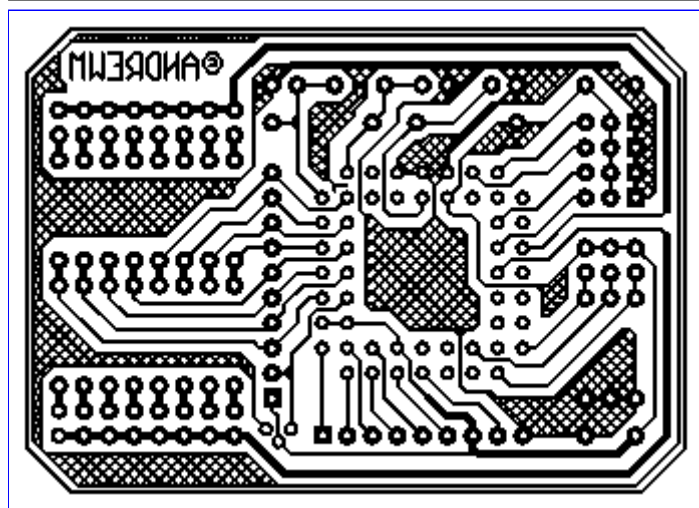
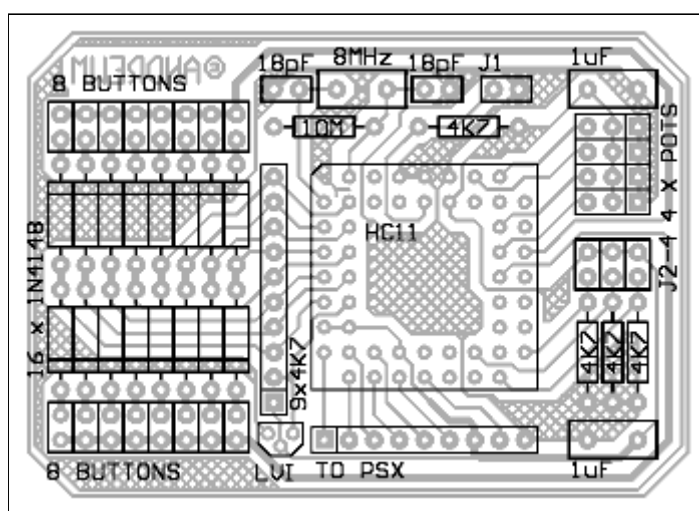
电路板下面的九个插座连到PS主机给微处理器程序使用。各脚作用如下。

Pin #	作用
1	SCI RX FOR RS-232 comms (not used by psx)
2	SCI TX "

3	DATA (pin 1 on PSX)
4	CLOCK (pin 2 on PSX)
5	COMMAND (pin 7 on PSX)
6	ATT (pin 6 on PSX)
7	VCC (pin 5 on PSX)
8	ACK (pin 9 on PSX)
9	GND (pin 4 on PSX)

最后在电路板下部的LVI (low voltage inhibitor) 作用是保持HC11 在复位状态当没足够电压安全运行时。如果你找不到可以不用不影响电路工作但可能导至过流烧保险丝。

电路板



上面电路板所用元件的编号及型号如下。

元件表

U1	MC68HC11E2
U2	MC 34064 Low voltage inhibitor
XTAL	8Mhz 晶振

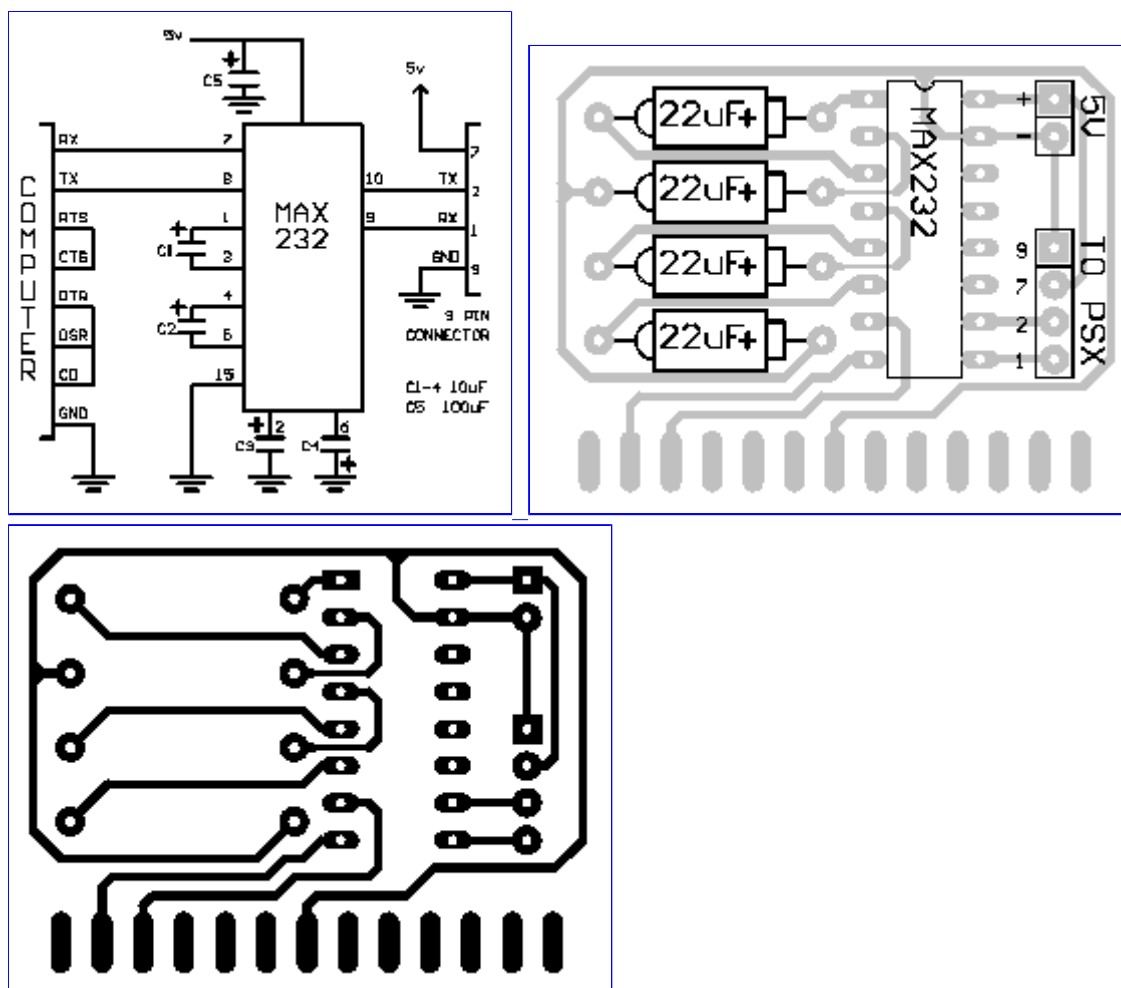
C1, C2	18pF 瓷片电容
C3, C4	1uF mono or MKT 电容
R1	10M 电阻
R2- R5	4K7 电阻
RP1	4K7 x 9 电阻排
CON1	9 脚插座
D1- D16	1N4148 二极管
J1	引导模式跳线
J2- J4	手制模式跳线
POT1- 4	模拟输入
Button 1- 16	16路跳线开关

软件

软件包包括电路板PCB 图(auto/easytrax 格式) 这里下载[PSXCONT.ZIP](#)

HC11 的下载软件

程序包括在ZIP文件中，文件名EELoader.EXE。这是一个IBM 执行文件用作下载代码进HC11。九针接口的第一、二脚是一个5V RS232 口，可以通过如下一个MAX232转换器连接到任何IBM 兼容PC的串口上。



使用它连接到你的PC串口，打EELoad PSXCONT.S19 /Cx，x 是使用的串口，接着按屏幕上的指令做。

© 1998 Andrew J McCubbin

Maintained by andrewm@quicknet.com.au