



SKPS Rev 2.0

PS2 Controller Starter Kit



User's Manual

V1.1

March 2016

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Cytron Technologies Incorporated with respect to the accuracy or use of such information or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Cytron Technologies's products as critical components in life support systems is not authorized except with express written approval by Cytron Technologies. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Index

1.	Introduction and Overview	4
2.	Packing List	5
3.	Product Specification and Limitations	6
4.	Board and Product Layout	7
5.	Hardware Interface	8
5.1	Interface with microcontroller	9
5.2	Interface with computer	11
5.3	PS2 Controller	17
6.	Protocol	22
6.1	Non-ASCII mode (microcontroller)	22
6.1.1	PS2 Button and joystick status	22
6.1.2	On Board vibrator motor control	23
6.1.3	All PS2 button and joystick status	24
6.2	ASCII mode (computer)	26
6.2.1	PS2 button and joystick status	26
6.2.2	Read key functions	27
6.2.3	PS2 vibrator motor control	28
7.	Getting Started	29
8.	Warranty	32
9.	Appendices	33

Document Version Change Log

No.	Doc Version	Date	Remarks
1	V1.0	Dec 14	1st release
2	V1.1	Mar 16	Add Change Log page
3	V1.1	Mar 16	Remove “SONY” for recommended PS2 controller

1.0 INTRODUCTION AND OVERVIEW

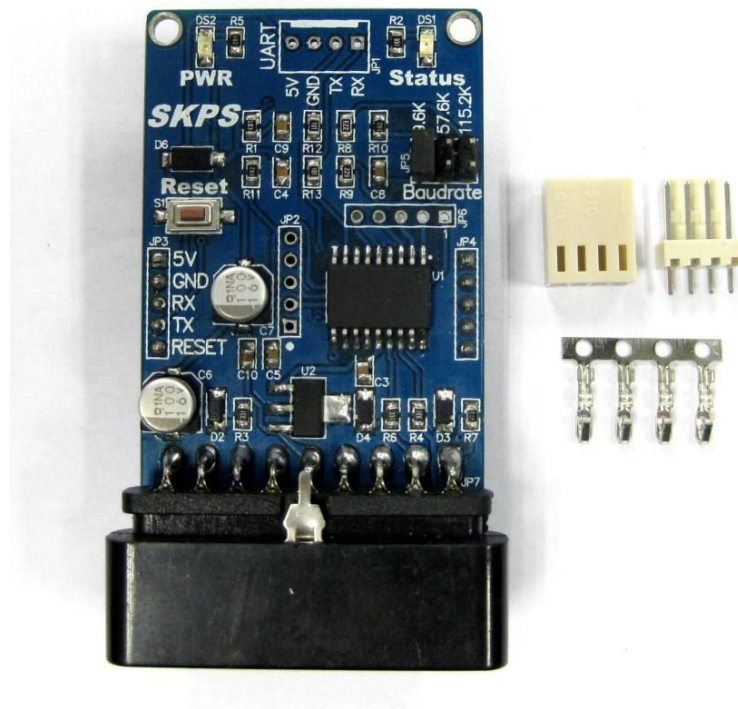
SONY Play Station 2 (PS2) controller is relatively easy to obtain from any game store and it offers good human interface with proper gesture to hold and press button, suitable for control system. More and more developers are looking into applying existing PS2 controller to control system. The major problem to archive this is the socket for PS2 and the protocol to communicate with it. PS2 uses special protocol to obtain the status (digital and analog) of each button and analog stick on PS2 controller.

Thus, Cytron Technologies has designed a PS2 Controller Starter Kit, Revision 2.0 offering better solution. This starter kit offers a compact yet reliable PS2 Controller Converter for user. SKPS's power and UART pin is compatible with SKKCA and [SKXBee](#). A host (computer or microcontroller) is needed to communicate with SKPS through UART. Reading Joy-stick button's state will be as easy as reading data via UART. It offers a standard connector for SONY PS2 controller to plug-in, wired or wireless

Features:

- 5V powered, low current consumption, less than 150mA if vibrator motor is not activated.
- 5V TTL UART interface to microcontroller or computer.
- Simple to use UART protocol, microcontroller or ASCII mode.
- Vibrator motor on PS2 is controllable.
- Wired and Wireless PS2 controller is supported.
- PS2 Controller will automatic operate in analog mode.
- An indicator LED act as indicator.
- Dimension 7.4cm x 3.9cm

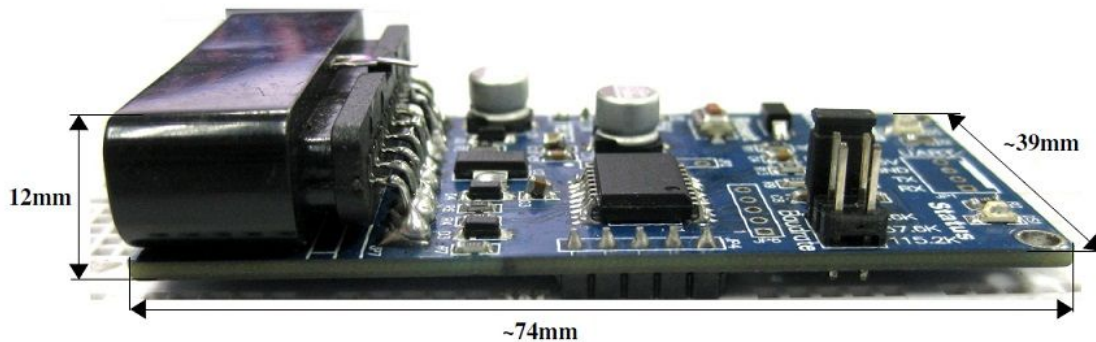
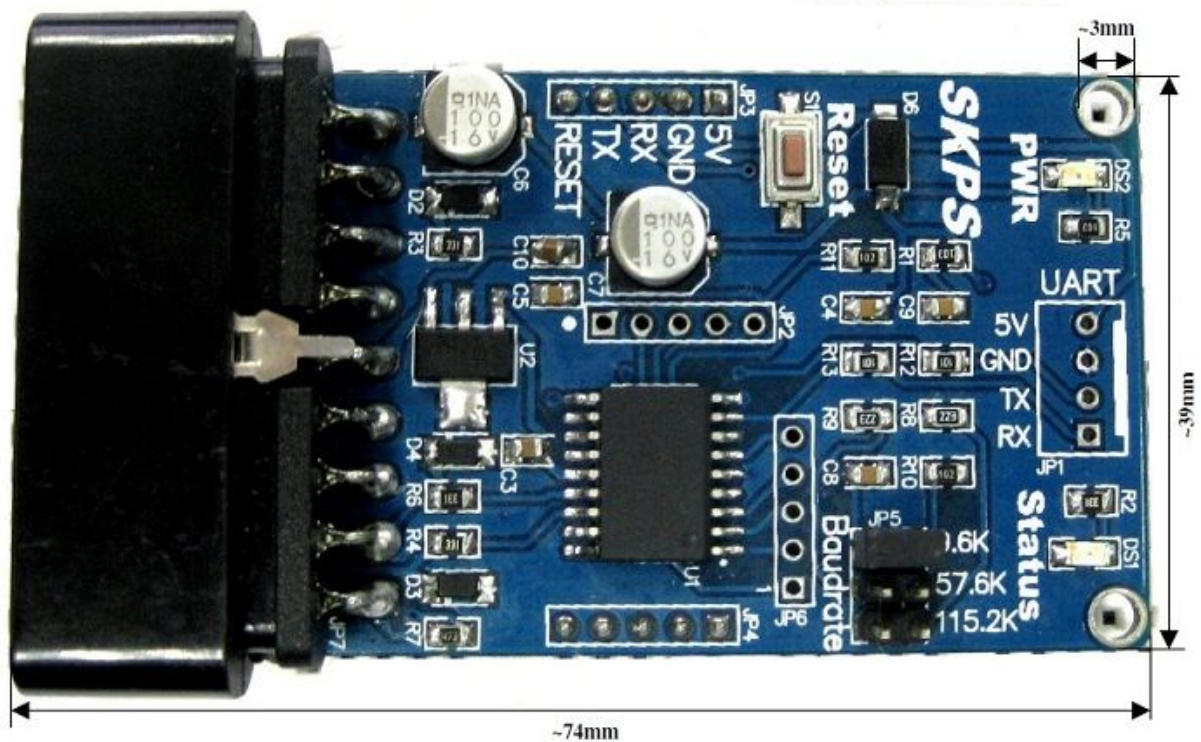
2.0 PACKING LIST



1. 1 x [SKPS Rev2.0](#)
2. 1 x [2510 4 ways connector](#) with [terminal pin](#).
3. User's manual, and sample source code can be downloaded from <http://www.cytron.com.my>.

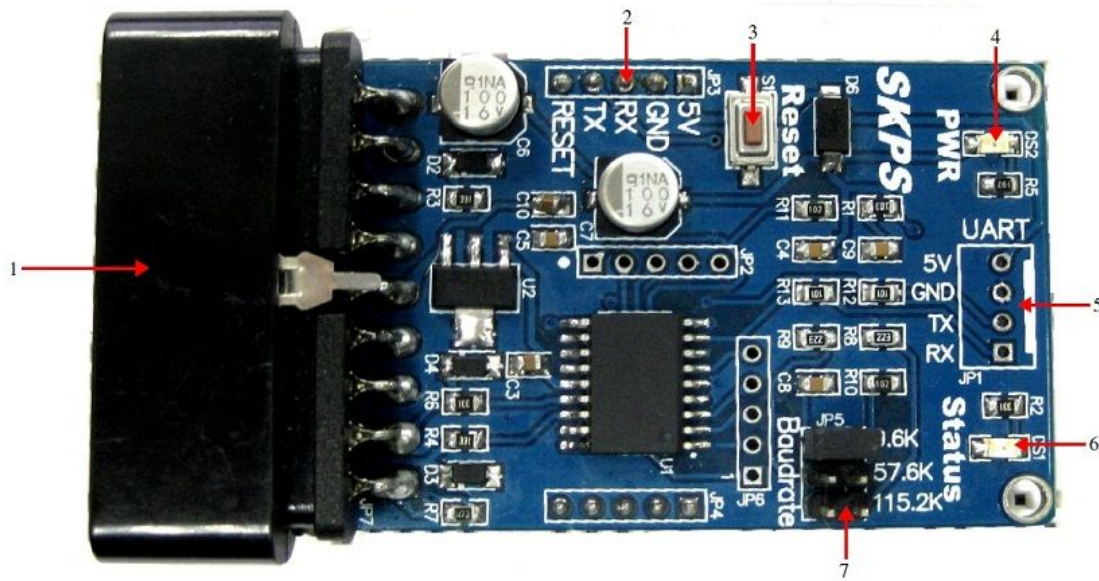
3.0 PRODUCT SPECIFICATION AND LIMITATIONS

Dimensions



No	Parameters	Min	Typical	Max	Unit
1	Input Voltage (Logic Operation Voltage)	4.5	-	5.5	V
2	Current Consumption	100	150	500	mA
3	V_{IOH} (Logic Input – High Level)	3.5	-	5.5	V
4	V_{IOL} (Logic Input – Low Level)	0	0	1.0	V

4.0 PRODUCT LAYOUT



Components on SKPS and their functions:

1. SONY PS2 Connector Socket – Please connect PS2 Controller plug here, wireless or wired.
2. 5 ways header pin - For external power and UART interface to microcontroller. Reset pin of on board controller is also one of it.
3. On board reset button. SKPS will require around 30ms to ready after reset.
4. Power indicator LED (Green) – Power indicator LED. If power is supply correctly, this LED will illuminate.
5. 4 ways 2510 pad – Optional connection to UC00A/B or UART device. You may solder 4 ways 2510 straight or right angle connector.
6. Status indicator LED (Orange) - This LED will blink and illuminate with different brightness depending on the PS2 status.

LED Condition	Description
Blink with different brightness	Once power up, if there is no SONY PS2 controller connected or detected..
Stay illuminated with low brightness	SONY PS2 controller detected and communication is working fine.
Stay illuminated with high brightness	Digital button on SONY PS2 controller is pressed.

7. Baud Rate selector - To select the preferable UART baud rate. For SKPS to update the obtain latest baudrate from selector, SKPS need to be reset or re-apply power.

5.0 HARDWARE INTERFACE

Generally, there are 2 methods of using SKPS Rev 2.0, with microcontroller or computer (normally for functionality test). Both methods will use the [UART](#) (Universal Asynchronous Receiver and Transmitter).

5-way site header pin functions:

Label	Definition	Function
5V	Power Input for SKPS	External power source for SKPS, the typical voltage is 5V. Please ensure this 5V source is a stable and regulated supply. Please do not use normal AC-DC adaptor to power it. It is recommended to use linear regulator (7805) to provide 5V supply.
GND	Ground or negative	Ground of power and signal.
RX	SKPS UART Receive signal	This is SKPS's UART receiver pin, it should be interfaced to 5V logic UART, no divider is necessary. This is an input pin to SKPS. It should be connected to microcontroller's UART transmitter pin.
TX	SKPS UART Transmit signal	This is SKPS's UART transmitter pin; it should be interfaced to 5V logic UART. This is an output pin from SKPS. It should be connected to microcontroller's UART receiver pin.
RESET	SKPS Reset pin	Reset pin of SKPS. Optional for user to use. It is same as on board push button. If you want to reset the SKPS from external, this pin should be connected to a push button (another terminal of button to GND), or to collector terminal of a NPN transistor. Please provide 30 ms (milli second) of time for SKPS to be ready after reset.

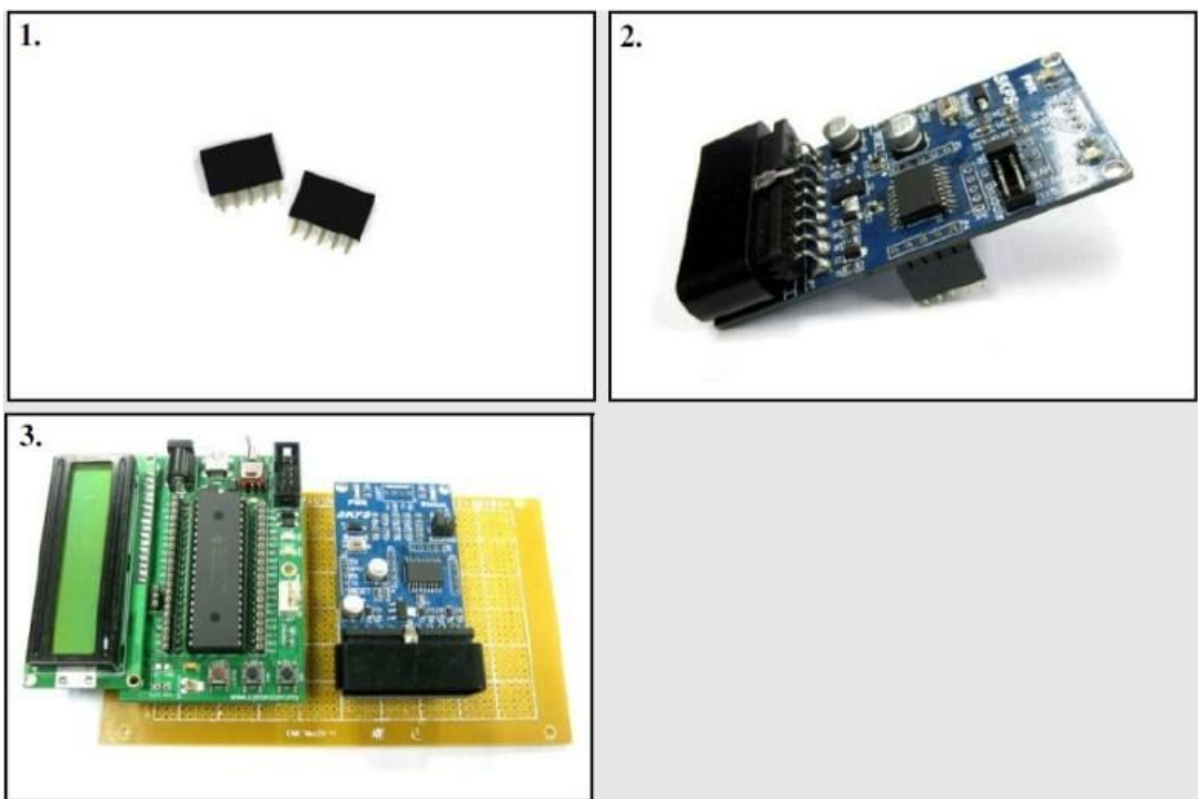
NOTE:

1. SKPS can only be powered by either by PC through 4 way 2510 connector or external power (5V). The 5V must be a stable supply. Any ripple higher than 5.5V will spoil the controller on SKPS and it is not replaceable.
2. Only left 5-way header pad (which is being labeled) have functions. The right 5-way header pad is dummy (no function) which is designed to support SKPS.

5.1 Interface with Microcontroller

There is no constrain of which brand of type of microcontroller to be use to interface with [SKPS Rev 2.0](#). As long as the microcontroller can interface/communicate with 5V TTL UART. Most of microcontroller come with UART peripheral such as [PIC16F877A](#), [876A](#), [887](#), PIC18F452, [4520](#), [4550](#), [2550](#), and this include Arduino. This manual will show the interface with [SK40C](#) with [PIC16F887](#) (running internal oscillator of 8MHz), sample code can be obtained from www.cytron.com.my

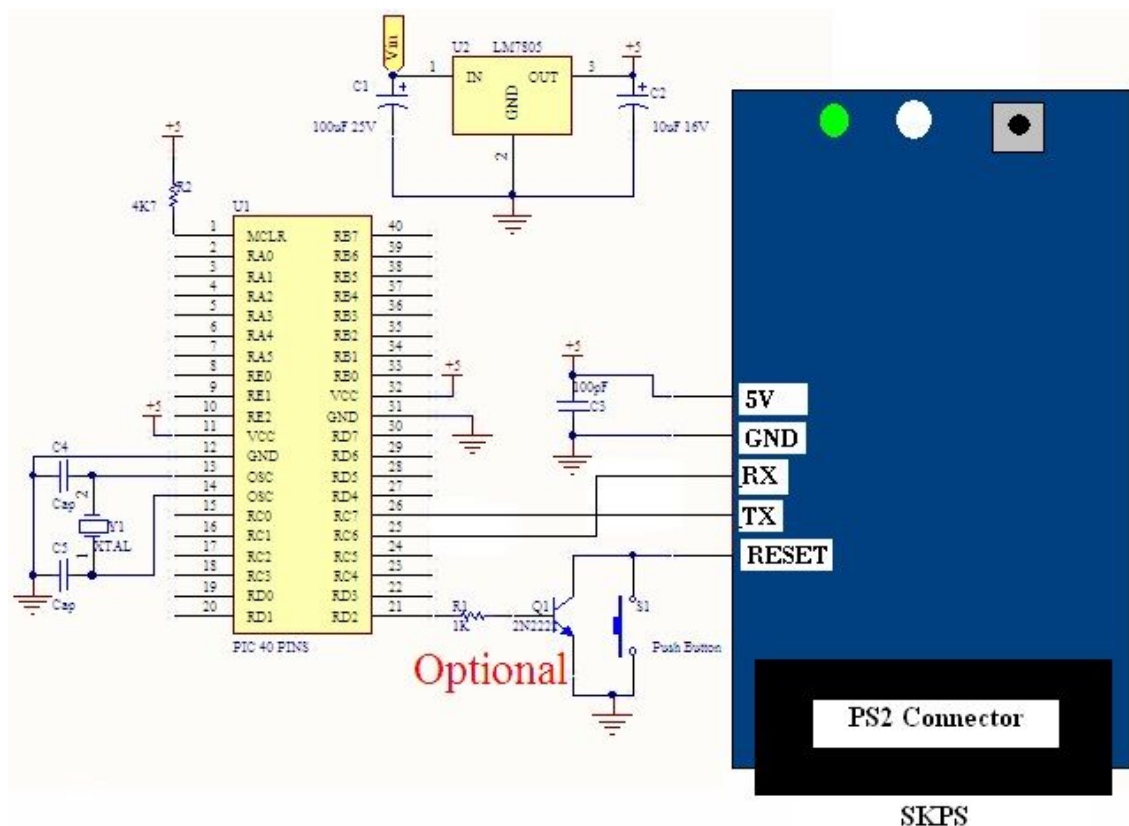
1. To begin, user may get [1x10 straight female header](#), cut it to 5-way. You will need two. Plug both the female header to the SKPS header pin, and place it properly on PCB board/donut board. Solder it and connect the necessary pin. Simplest to start is to supply 5V from voltage regulator of [SK40C](#).



2. Once the SKPS is powered, the green LED will illuminate and orange LED will blinking with different brightness. Connect SONY PS2 controller to the socket and orange LED should stop blinking and maintain illuminate with low brightness. With this response/condition, is shows SKPS Rev2.0 is working and it can communicate with SONY PS2 controller. You can further press any digital button on PS2 and observe the orange LED, whenever a digital button is press, the LED will illuminate with full brightness.
3. Now, in order to use the information from SKPS, microcontroller is needed. You must connect the UART pins. Microcontroller RX pin must be connected to SKPS's TX pin; on the other hand, Microcontroller TX pin, connect to SKPS's RX pin. This is cross connect. For 40-pin PIC on SK40C, RX pin of PIC is located at RC7, and TX

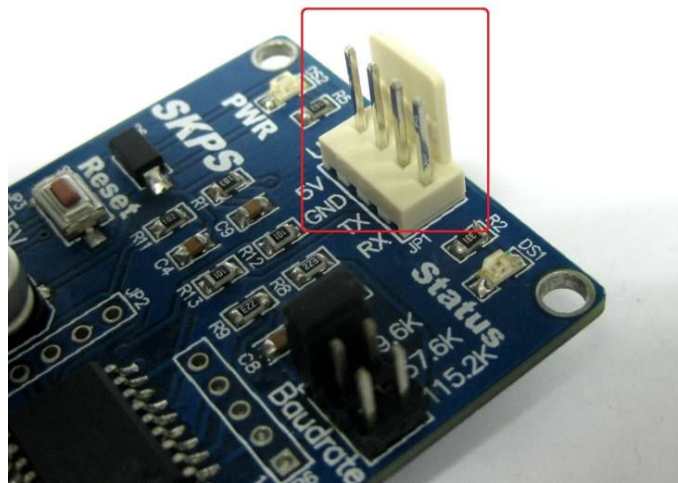
pin is RC6. No extra component is needed, just simple direct wire is sufficient.

4. Once 5V, GND, TX and RX pin is connected, your microcontroller is ready to communicate with SKPS and this require software development on that particular PIC. You can always download the sample source code and modify it according to your need.
5. Finally, the RESET pin of SKPS. This is an **optional** pin for user to connect as the basic required pins are 5V, GND, TX and RX; there is also a RESET button on SKPS Rev 2.0 board. However, this RESET pin offer option for microcontroller to reset the SKPS. If you want the microcontroller to be able to reset SKPS, you can refer to the following schematic. Push button is optional. Please be reminded, after Reset, SKPS will require ~30ms to be ready for communciation.

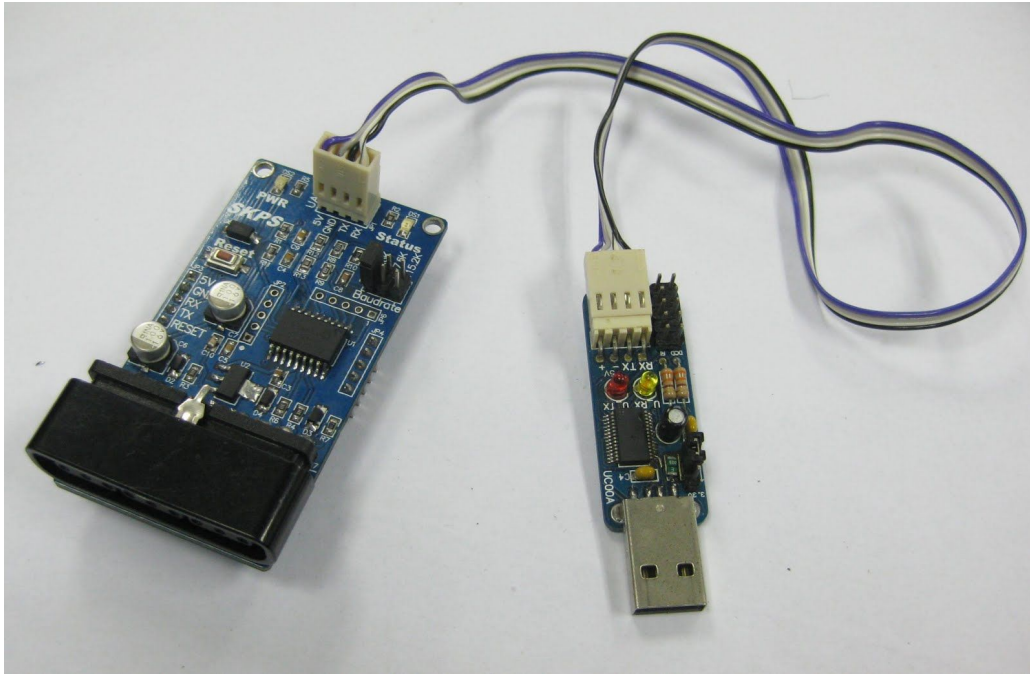


5.2 Interface with Computer

1. Using computer to interface with SKPS is much easier because the keyboard and monitor is ready. Nonetheless, you will still need to connect SKPS to computer. SKPS uses UART to send/receive information to/from host. Computer does not have a UART interface. Yet, USB provide the bridge to get UART. You can use [UC00A](#) or [UC00B](#) for that purpose.
2. To begin, this manual will show the interface using [UC00A](#). Again, there are several methods to do this. The better way is to use the 4-way 2510 connector provided and solder it on SKPS. As shown.



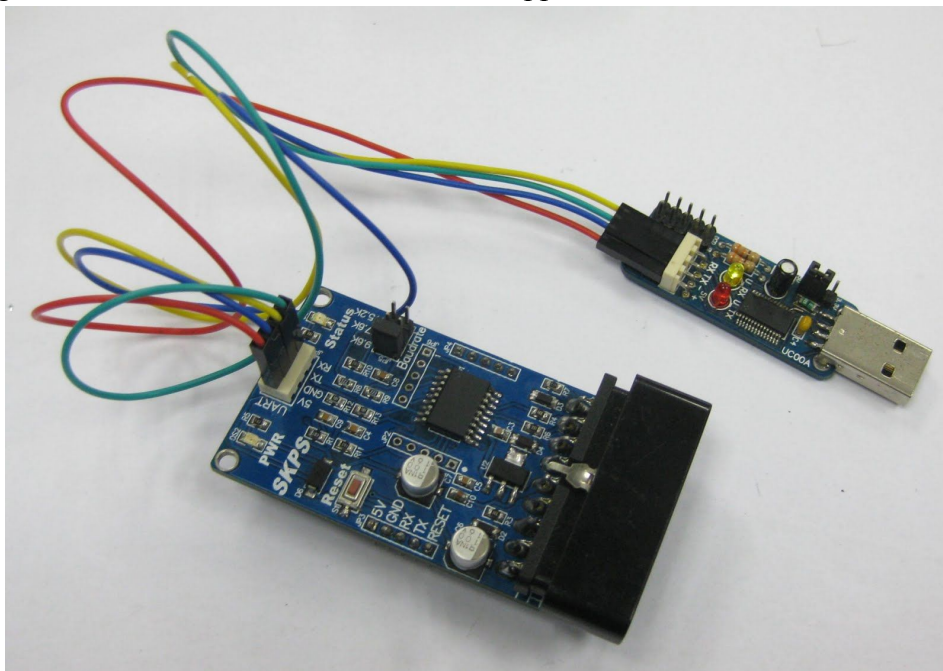
3. Create a 4 ways cable using rainbow wire, connect it to UC00A as shown. User may refer UC00A User's Manual to create 4 ways cable. Do take note that TX and RX line of UC00A should be cross connected to SKPS.



4. Or, another simpler but not really permanent method is to use [Female to Female jumper wires](#). Connect it in following configuration:

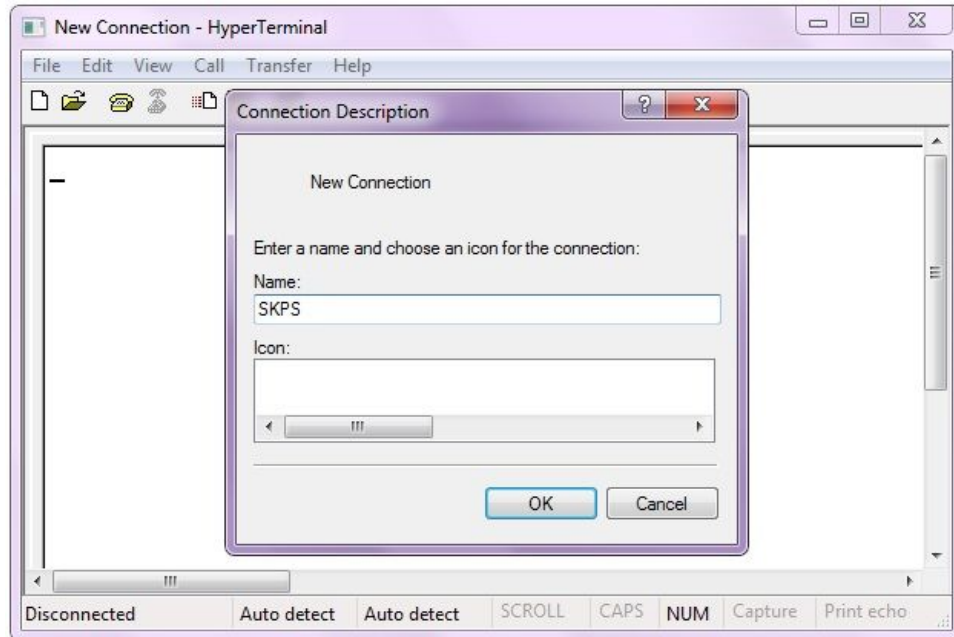
SKPS	UC00A
5V	+
GND	-
TX	RX
RX	TX

5. Again, do take note that RX and TX is swapped.

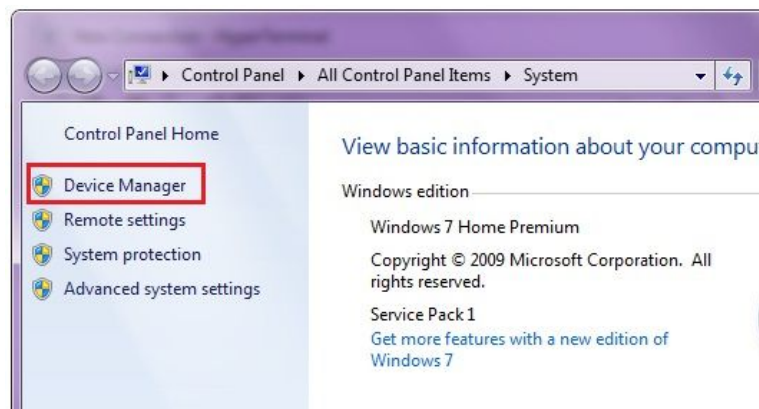


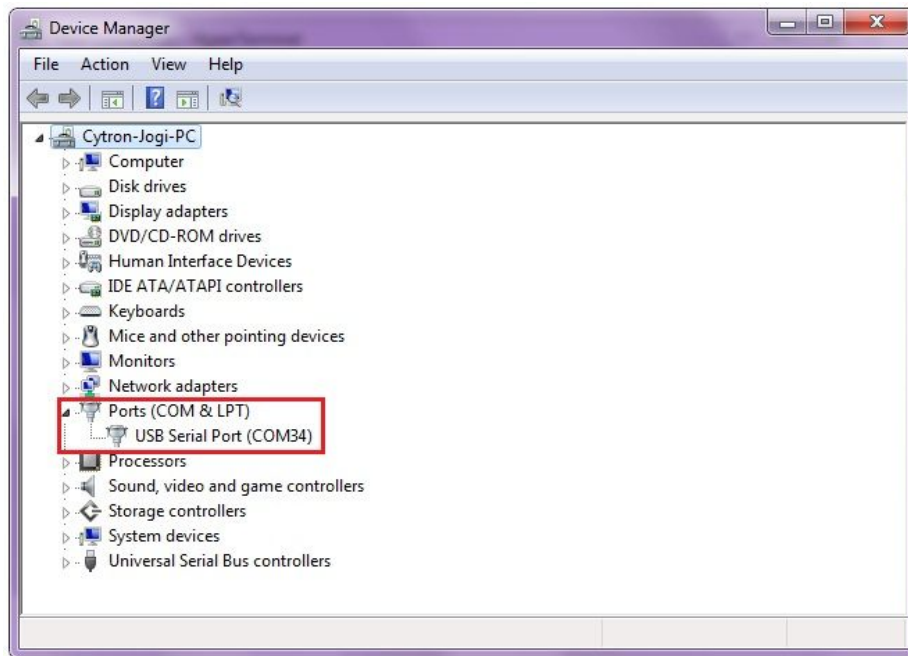
6. Connect SONY PS2 controller to the socket. Connect the UC00A to computer's USB port and if this is 1st time you UC00A, you will need to install the driver. Please download the driver from [here](#) and refer to the [Installation Guide](#).

7. While installing the driver, you can try the SKPS response by press any button on PS2 and the Orange LED will illuminate with full brightness.
8. After plug in the UC00A to computer and installation of driver, user is ready to test the functionality of SKPS.
9. Open the HyperTerminal. Enter a name and choose an icon for connection as picture below then click OK.

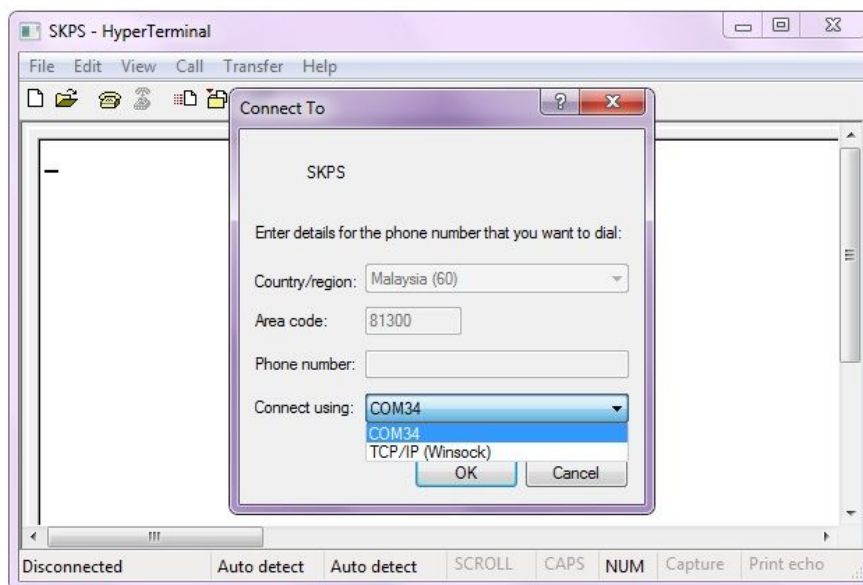


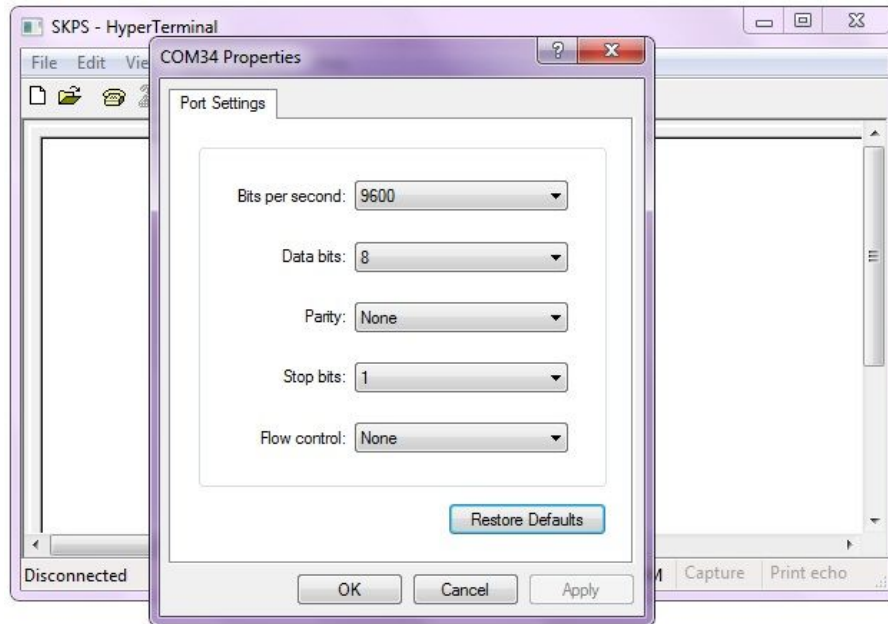
10. Connect using USB Serial Port. If you are not sure which COM is it, please follow step (11) to (12).
11. Go to Start, right click on My Computer and choose Properties.
12. Click on Device Manager. Device Manager Table will show out. At Device Manager Table, choose Ports (COM & LPT) and you can check your USB Serial Port COM. The Serial Port COM is “COM34” in this case.



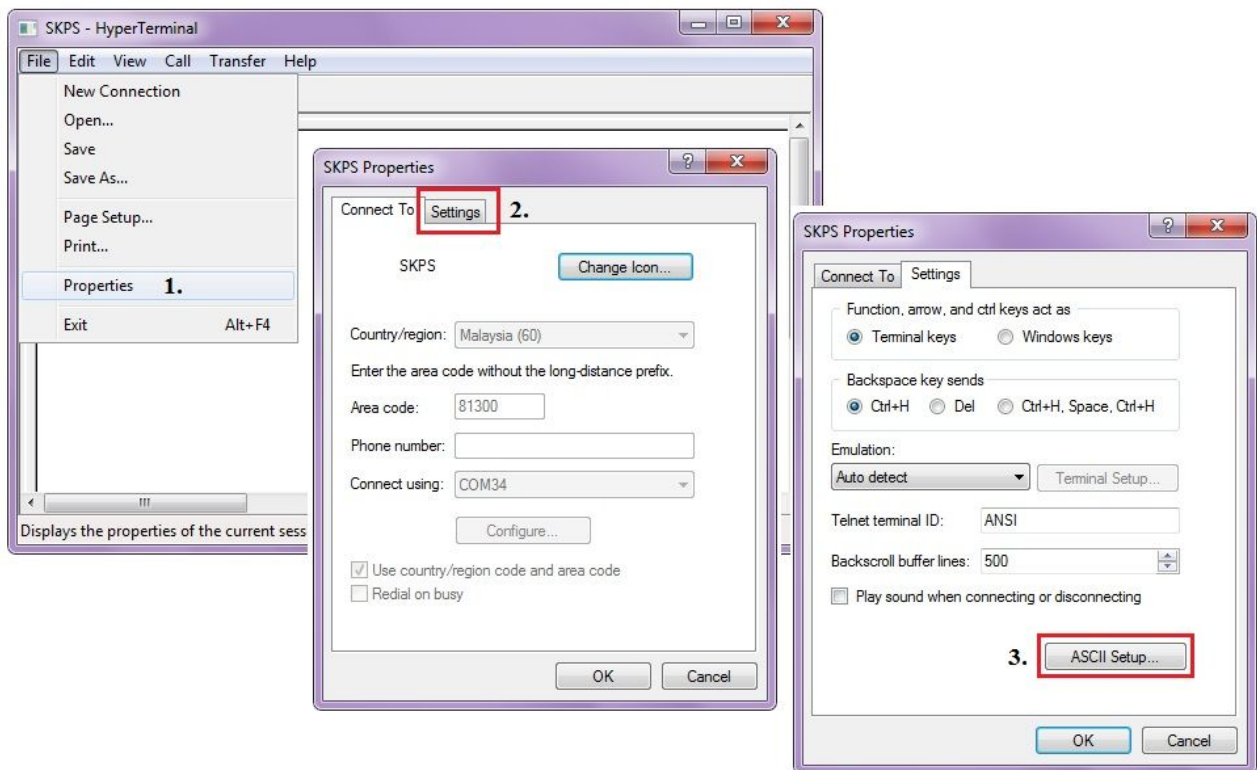


13. Set the Port Setting as picture below. Bits per second must be same with SKPS Baud Rate and Flow control must be set to none. After finish setting, click Apply and then click OK.

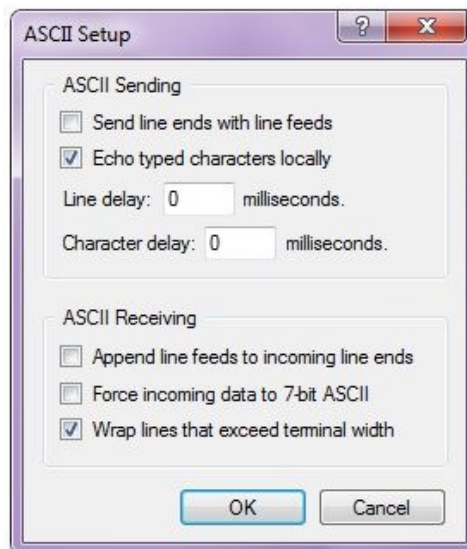




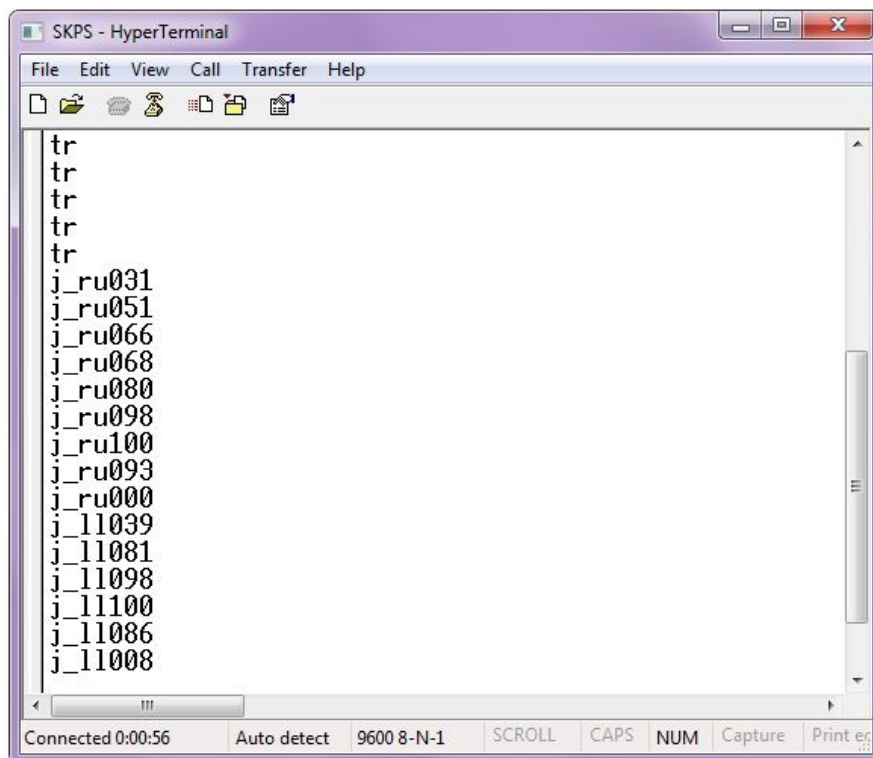
14. Go to File and select Properties. 'skps' Properties table will show. Choose Setting tab and click ASCII Setup tab.



15. On ASCII Setup, click on Echo type characters locally and then click OK.



16. After all settings are complete, user may now check the functionality of SKPS. Please refer ASCII Mode (PC) protocol in Section 6.2. Type “ae” on keyboard and press “enter” to activate auto-response mode. SKPS will update HyperTerminal with any key pressed or changes of joystick automatically. Type “ad” and “enter” to disable this feature.



NOTE:

Besides HyperTerminal, X-CTU, Serial Monitor on Arduino or any Terminal program can be used to obtain response from SKPS.

5.3 Choose PS2 Controller

User are free to choose any type of PS2 controller in the market, either wired or wireless. Figure below shows example of wired and wireless type of PS2 controller. There are many types of PS controller in the market and the sensitivity for each type also different. User is advised to use original PS2 controller. Cytron Technologies does not guarantee compatibility for all PS2 controllers. No modification is needed to connect to SKPS.



Example PS2 Controller: Wired SONY PS2 Controller

RF Wireless Joypad Controller for PS / PS2



Example PS2 Controller: Wireless SONY PS2 Controller

Figure below shows the digital buttons and analog joystick on a typical PS2 controller.



The analog value of Left Joystick and Right Joystick can be read from SKPS too. Each joystick have 2 axes, and there are two formats of output. These formats will be explained later. User may choose format 1 or format 2.

NOTE:

SONY PS2 controller does not come with SKPS, please purchase separately from Cytron Technologies website. It is advised to use PS2 controller from Cytron Technologies because all PS2 controller is tested before it is being shipped to customer.

Analog output format 1:

There are 2 variables for each joystick, axis X and axis Y, this is what we commonly use in graph drawing. On PS2 joystick, as example, when user push the joystick up or down, the Y axis will change. Meanwhile if user push the joystick left or right, the value of X axis will change.

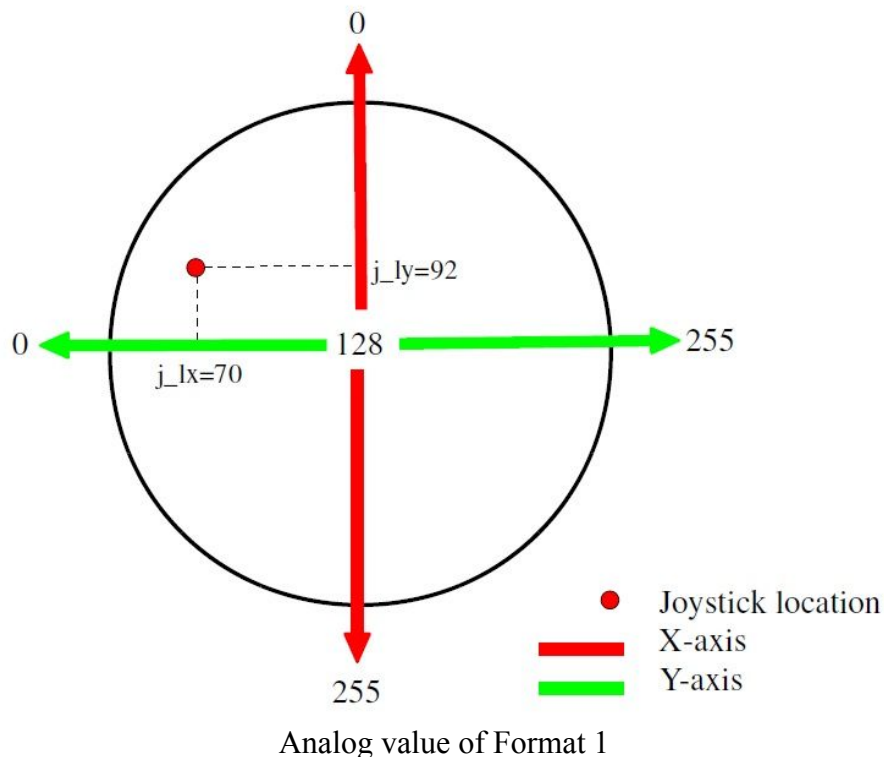
Y Axis:

- Middle (neutral), value is 128
- Push up, value change from 128 to 0
- Push down, value change from 128 to 255.

X Axis:

- Middle(neutral, value is 128
- Push towards left, value change from 128 to 0
- Push towards right, value change from 128 to 255

Left joystick have X axis and Y axis, Right joystick also have its own X and Y axis.



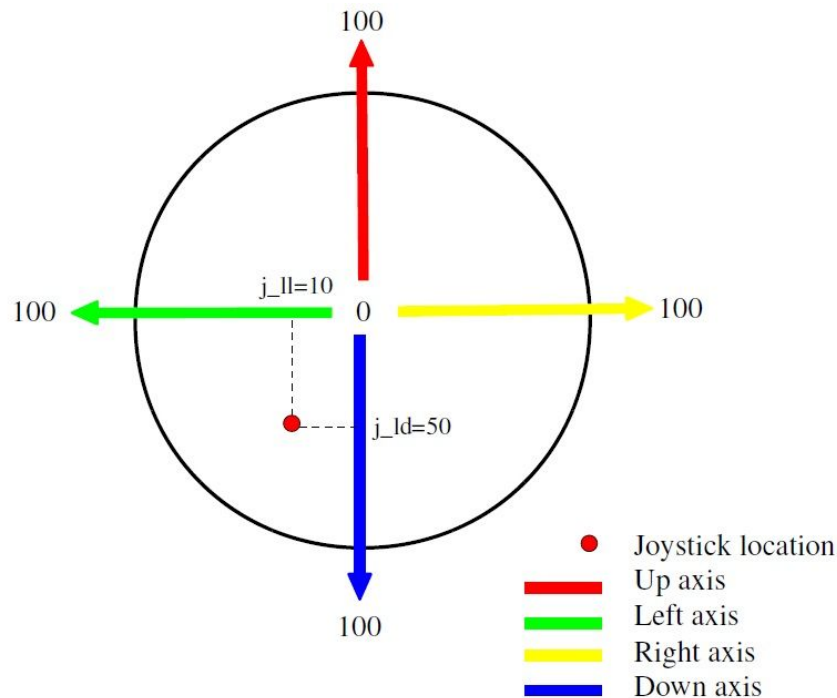
Taking an example, let's say the left joystick is being push to the position shown in the figure. It is being push to left top corner. The value of left joystick will change will should get approximately:

- $j_{lx} = 70$
- $j_{ly} = 92$

This is just an example, we notice a lot of PS2 joystick do not provide linear analog value across the range. There is a big dead zone near the middle position where analog value does not change. User is require to “play” around with the joystick.

Analog output format 2:

In format 2, there are four variables for each joystick. The four variables are up, down, left and right. When user move the joystick in any direction, the value of these four variables change from 0 to 100. Left and Right joystick will have 4 independent variables.



Analog value of Format 2

Taking an example, let's say the left joystick is being push to the position shown in the figure. It is being push to left down corner. The value of left joystick will change will should get approximately:

- $j_ll = 10$
- $j_ld = 50$

You can choose which ever format to use by sending particular command to SKPS.

Please do take note that the the format 1 variables are:

- j_lx - Left joystick, X axis
- j_ly - Left joystick, Y axis
- j_rx - Right joystick, X axis
- j_ry - Right joystick, Y axis

Format 2 variables are:

- j_lu - Left joystick, up axis
- j_ld - Left joystick, down axis
- j_ll - Left joystick, left axis
- j_lr - Left joystick, right axis
- j_ru - Right joystick, up axis
- j_rd - Right joystick, down axis
- j_rl - Right joystick, left axis
- j_rr - Right joystick, right axis

j_XX



l - mean Left joystick r - mean Right joystick	x - mean X axis, format 1, range 0 to 255 y - mean Y axis, format 1, range 0 to 255 u - mean Up axis, format 2, range 0 to 100 d - mean Down axis, format 2, range 0 to 100 l - mean Left axis, format 2, range 0 to 100 r - mean Right axis, format 2, range 0 to 100
---	---

6. PROTOCOL

Generally, there are 2 protocol modes, Non-ASCII mode (Microcontroller) and ASCII mode (computer). Do not get confused, both mode can be use anytime, just if SKPS is connected to microcontroller, user will prefer to use non-ASCII mode because it is easier to write program. On the other hand, if SKPS is conected to computer, ACSII mode is easier. All command and response is through UART communication.

6.1 Non-ASCII Mode (Microcontroller)

By default, SKPS is in passive condition, where it waits for command from UART. If microcontroller send a command, it will response based on the command and the status of PS2 button and joystick. The command is simple, it is in value. Each value will request SKPS to check the particular button or joystick status on PS2 and response.

6.1.1 PS2 Button and Joystick Status

Send (decimal)	Button on PS2	Description
0	Select button	<p>Applied to value 0 to 15 (decimal): SKPS will return the status of corresponding button when the particular decimal value is received 0 if the button is pressed 1 if button is not pressed</p> <p>Example: If microcontroller sent 10 (decimal), SKPS will check L1 button status on PS2 Controller. SKPS will return: 0 if L1 button is pressed or 1 if L1 is not pressed.</p>
1	left joystick centre button	
2	right joystick centre button	
3	START button	
4	up button	
5	right button	
6	down button	
7	left button	
8	L2 button	
9	R2 button	
10	L1 button	
11	R1 button	
12	triangle button	
13	circle button	
14		

15	cross button square button	
16	left joystick x-axis	<p>Applied to value from 16 to 27 (decimal): SKPS will return the particular value of corresponding joystick in the selected axis</p> <p>Example : If microcontroller sent 20 (decimal), SKPS will read and return the value of left joystick up.</p>
17	left joystick y-axis	
18	right joystick x-axis	
19	right joystick y-axis	
20	left joystick up value	
21	left joystick down value	
22	left joystick left value	
23	left joystick right value	
24	right joystick up value	
25	right joystick down value	
26	right joystick left value	
27	right joystick right value	
28	Return the connection status of PS2 and SKPS	SKPS will read controller status 1 is return if controller is detected

6.1.2 On board Vibrator Motor Control

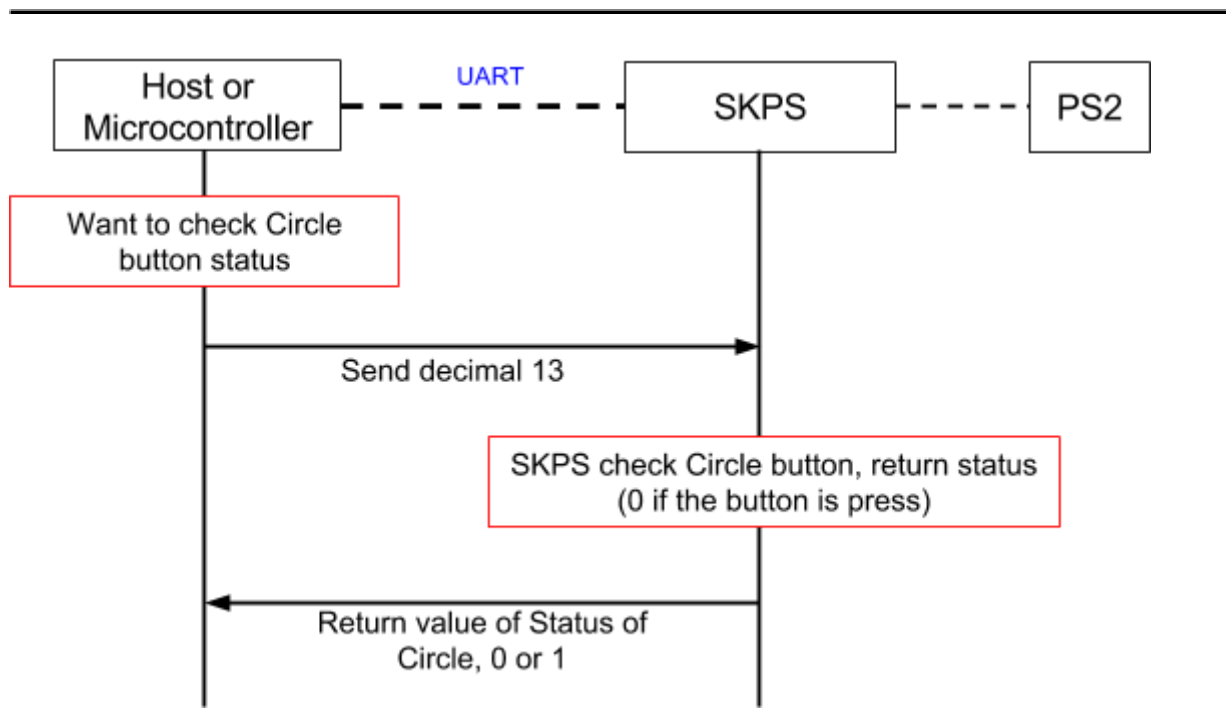
Following command require two bytes of data send from host/microcontroller. 1st byte to indicator which vibrator motor, following byte (2nd byte) to indicator motor status or the speed.

send 1st byte (decimal)	send 2nd byte (decimal)	Controller	Description
29	motor1 value	smaller vibrator on right	motor1 = 1 (motor on) motor1 = 0 (motor off)
30	motor2 value	bigger vibrator on left	motor2 = 0 to 255 (adjustable speed)

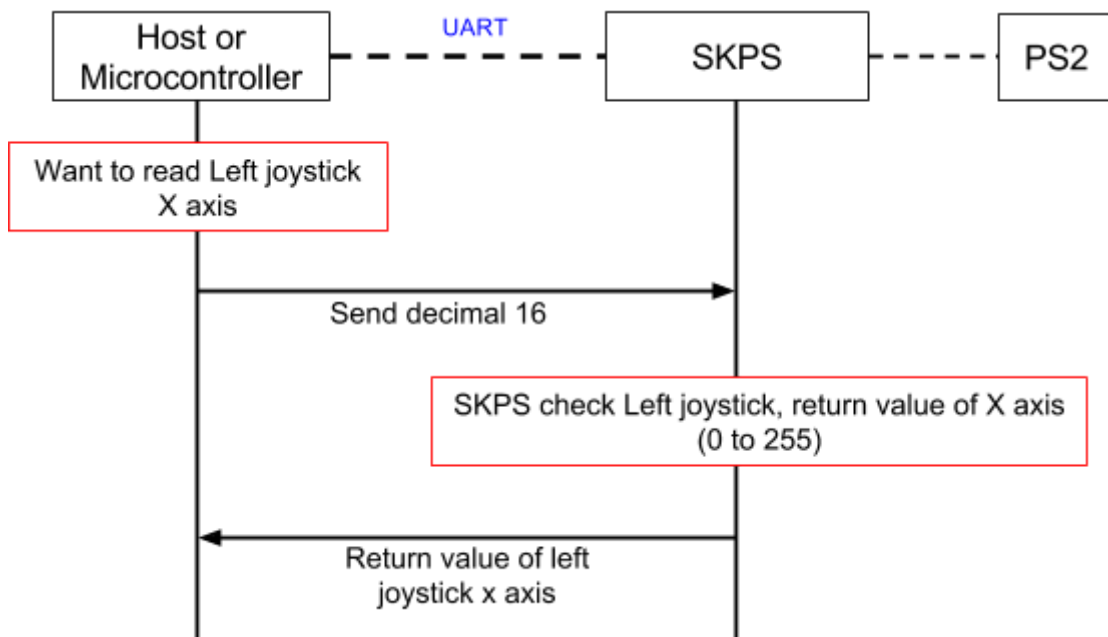
6.1.3 All PS2 Button and Joystick Status

Send (decimal)	Response from PS2	Description
31	<p>6 bytes of data: 1st byte: digital button group 1 2nd byte: digital button group 2 3rd byte: Right Joystick X axis 4th byte: Right Joystick Y axis 5th byte: Left Joystick X axis 6th byte: Left Joystick Y axis</p>	<p>SKPS will return the status of all digital and analog joystick of PS2</p> <p>Digital group 1: bit 7 (MSB): Left button bit 6: Down button bit 5: Right button bit 4: Up button bit 3: Start button bit 2: Right Joystick Center button bit 1: Left Joystick Center button bit 0 (LSB): Select button</p> <p>Digital group 2: bit 7 (MSB): Square button bit 6: Cross button bit 5: Circle button bit 4: Triangle button bit 3: R1 button bit 2: L1 button bit 1: R2 button bit 0: L2 button</p> <p>0 if the button is pressed 1 if button is not pressed</p> <p>Example: If microcontroller sent 31 (decimal), SKPS will check all the button and joystick status and return 6 bytes to microcontroller.</p>

Examples:



Example of UART communication shows microcontroller obtain the Circle button of PS2



Example of UART communication shows microcontroller obtain the Left joystick, X axis value of PS2

6.2 ASCII Mode (Computer)

ASCII mode simply mean host send ASCII character. Computer normally send ASCII code and display ASCII code, so the simplest and easiest way to verified the working of SKPS is through computer.

6.2.1 PS2 Button, Joystick and Status

Send 2 ASCII	Button, Joystick and Status of Controller	Description
“up”	up button	<p>SKPS will return the status of corresponding button when the particular ASCII value is received ‘0’ if the button is pressed ‘1’ if button is not pressed</p> <p>Example: If microcontroller sent ‘r2’ (ASCII), SKPS will check R2 button status on PS2 Controller. SKPS will return ‘0’ if R2 button is pressed or ‘1’ if R2 is not pressed. [“Enter” after every result]</p>
“dw”	down button	
“lf”	left button	
“rg”	right button	
“tr”	triangle button	
“cr”	circle button	
“sq”	cross button	
“ci”	square button	
“st”	start button	
“se”	select button	
“jl”	left joystick centre button	
“jr”	right joystick centre button	
“l1”	L1 button	
“l2”	L2 button	
“r1”	R1 button	
“r2”	R2 button	
“lx”	left joystick x-axis	<p>SKPS will return the particular value of corresponding joystick in the selected axis (3 byte, in ASCII).</p> <p>Example: ‘235’ or ‘029’. [“Enter” after every result]</p>
“ly”	left joystick y-axis	
“rx”	right joystick x-axis	
“ry”	right joystick y-axis	

“lu”	left joystick up value	SKPS will return the particular value of corresponding joystick in the selected axis (3 byte, in ASCII). Example: ‘039’ or ‘027’. [“Enter” after every result]
“ld”	left joystick down value	
“ll”	left joystick left value	
“lr”	left joystick right value	
“ru”	right joystick up value	
“rd”	right joystick down value	
“rl”	right joystick left value	
“rr”	right joystick right value	
“cs”		SKPS will read controller status ‘1’ is return if controller is detected [“Enter” after the status]
unknown		SKPS will return ‘x’ if unknown data is sent. [“Enter” after the ‘x’]

6.2.2 Read Key Functions

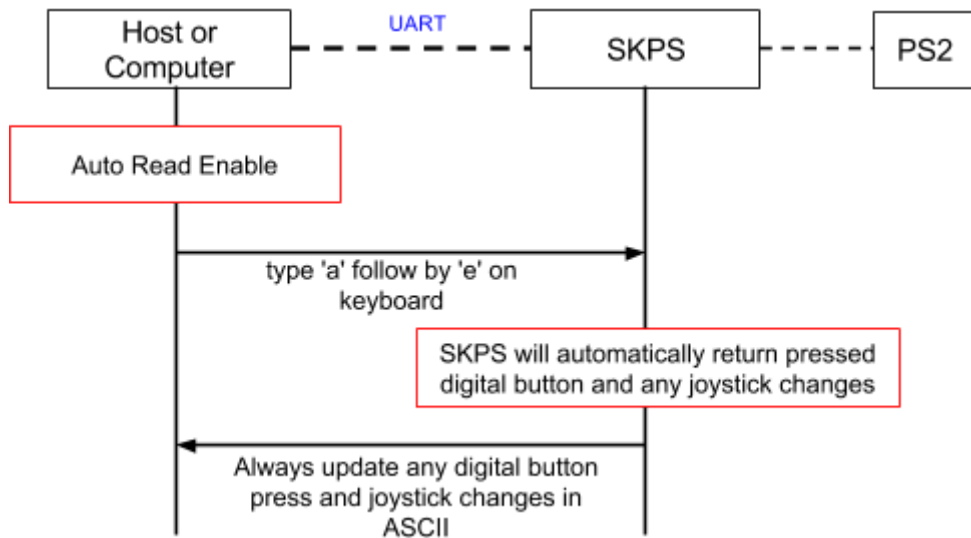
Send 2 ASCII	Computer will Receive	PS2	Description
“rk”	all key status	any press button/joystick value	SKPS will return pressed key and changed joystick value once, followed by “Enter” after result.
“ae”	auto read enable	any press button/joystick value	SKPS will enable return pressed key and changed joystick automatically, followed by “Enter” after result.
“ad”	auto read disable	any press button/joystick value	SKPS will disable return pressed key and changed joystick automatically, followed by “Enter”.

6.2.3 PS2 Vibrator Motor Control

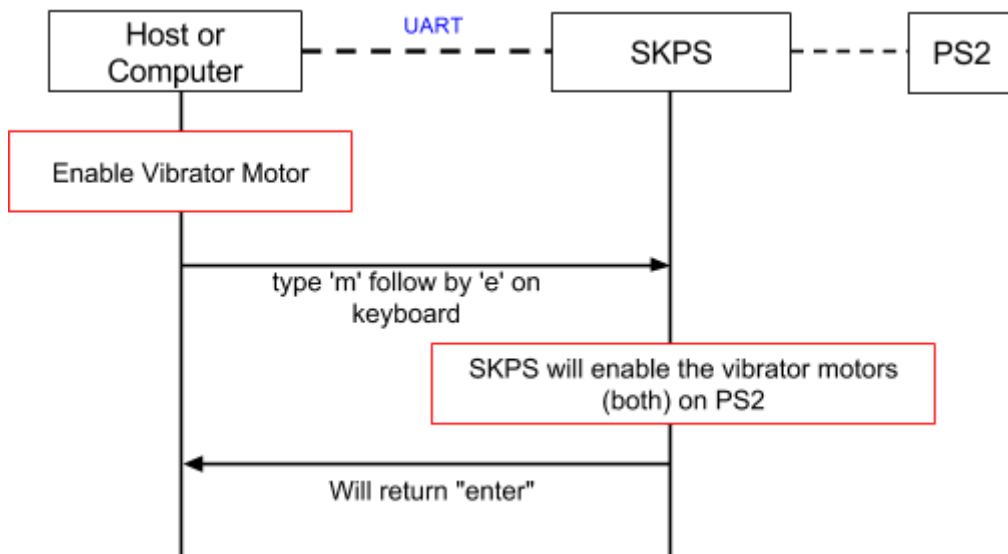
Send 2 ASCII	PS2	Description
--------------	-----	-------------

“me”	motor enable	SKPS will enable vibrator motor (both motor1 and motor2), followed by “Enter”.
“md”	motor disable	SKPS will disable vibrator motors (both motor), followed by “Enter”.

Example:



Example of enabling auto read operation



Example Communication of Motor Enable Operation

7.0 GETTING STARTED

For this section, SKPS will be interfaced with [PR23](#). Please refer PR23, DIY project from Cytron website for details example of interfacing to SKPS. PR23 shows the method of using SKXBee, since SKPS UART pin is designed to be compatible to SKXBee, user may replace SKXBee with SKPS on PR23. Please refer to the [details description of PR23](#). For method of writing most simple program to use with SKPS, please refer steps below and for sample full version source code, user may get it from Cytron Website ([same page as SKPS](#)).

- a. Set the configuration for UART. Make sure the baud rate is correct.

```
//setup USART
SPBRG = 0x81;           //set baud rate to 9600 for 20Mhz
BRGH = 1;              //baud rate high speed option
TXEN = 1;              //enable transmission
TX9 = 0;
CREN = 1;              //enable reception
SPEN = 1;              //enable serial port
RX9 = 0;
RCIE = 0;              //disable interrupt on eachdata received
```

- b. Definition of every button and joystick. For following example, “p_” is added in front of PS2 controller label for easy understandable and avoid crash with other variable or instructions.

```
//skps protocol
#define p_select        0
#define p_joy1         1
#define p_joyr         2
#define p_start        3
#define p_up           4
#define p_right        5
#define p_down         6
#define p_left         7
#define p_l2           8
#define p_r2           9
#define p_l1          10
#define p_rl          11
#define p_triangle    12
#define p_circle      13
#define p_cross       14
#define p_square      15
#define p_joy_lx      16
#define p_joy_ly      17
#define p_joy_rx      18
#define p_joy_ry      19
#define p_joy_lu      20
#define p_joy_ld      21
#define p_joy_ll      22
#define p_joy_lr      23
#define p_joy_ru      24
#define p_joy_rd      25
#define p_joy_rl      26
#define p_joy_rr      27

#define p_con_status  28
#define p_motor1     29
#define p_motor2     30
```

- c. Function to read SKPS and control the joystick. Below are examples of UART function and SKPS function. Comments explain each command. UART function controls process of sending and receiving data via UART while SKPS function is to read information on PS2 controller including function to read button and joystick and control the vibrator motor.

```
// uart function
//=====
void uart_send(unsigned char data) //function to send out a byte via uart
{
    while(TXIF==0); //wait for previous data to finish send out
    TXREG=data; //send new data
}

unsigned char uart_rec(void) //function to wait for a byte receive from uart
{
    unsigned char temp;
    while(RCIF==0); //wait for data to received
    temp=RCREG;
    return temp; //return the received data
}

// skps function
//=====
unsigned char skps(unsigned char data) //function to read button and joystick
//information on ps controller
{
    uart_send(data);
    return uart_rec();
}

unsigned char skps_vibrate(unsigned char motor, unsigned char value)
//function to control the vibrator motor
//on ps controller
{
    uart_send(motor);
    uart_send(value);
}
```

- d. Below is sample of using the above functions. For first command, when L1 is pressed, buzzer will beep. Second command is to test if there is any obstacle in front. For PR23 we use ultrasonic sensors to detect the obstacles. If obstacle is detected, motor in PS2 controller will vibrate.

```
//test button for horn
if(skps(p_l1)==0)buzzer=1; //if button L1 pressed, beep the buzzer
else buzzer=0;

//test if there is any obstacle in front
if(read_adc(CHANNEL0)<20) //if ultrasonic sensors detect obstacle in front,
{ //vibrate the ps controller
    skps_vibrate(p_motor2,255);
}
else skps_vibrate(p_motor2,0);
```

- e. After finish writing the program, user may compile it and load the source code. Please refer to PR23 on the method to download the code.
- f. Upon loading the program, PR23 is now ready to operate with SKPS. User may test it using PS2 controller, try each button and function. Have fun!

8.0 WARRANTY

- Product warranty is valid for 6 months.
- Warranty only applies to manufacturing defect.
- Damaged caused by miss-use is not covered under warranty
- Warranty does not cover freight cost for both ways.

Prepared by
Cytron Technologies Sdn. Bhd.
*16, Jalan Industri Ringan Kawasan Permatang Tinggi 2,
Kawasan Industri Ringan Permatang Tinggi,
14100 Simpang Ampat, Pulau Pinang, Malaysia.*

Tel: +604 - 504 1878

Fax: +604 - 504 0138

URL: www.cytron.com.my

Email: support@cytron.com.my

sales@cytron.com.my

9.0 APPENDICES

Notes Change:

- 30 March 2016:
 - Remove “Sony” from User manual. SKPS is compatible with any PS2 controller.
 - Update the revision of user manual