Playstation Servomotor Controller Interface

As of 2009, the PS2 has sold over 140 million units. With the PS2 still in production and so many controllers available, it's a natural to apply this controller for more than games. This makes it an excellent controller for servomotors and robotics.

The PS2 SMC (Servo Motor Controller) interface allows you to control six hobby servomotors using the Playstation controller (also called the Dual Shock 2 [DS2]). Both corded and cordless versions are supported. Therefore, the wireless SMC can be used for remote control of your moving platform.

by John Iovine

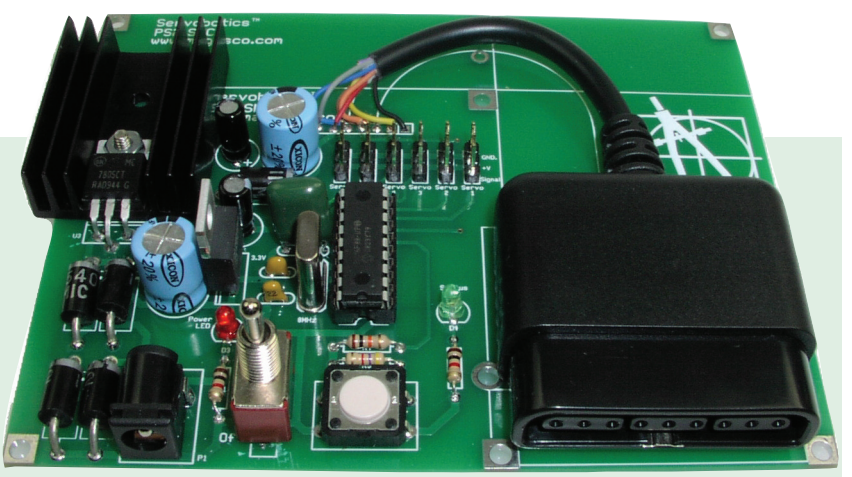


FIGURE 1. The PS2 interface controller.

The PS2's analog sticks give precise servomotor movements with speed control. Rumble feedback occurs when the servomotor is instructed to move out of range (stalling).

Sony has not released the communication protocol of its PlayStation controller, however, there are a few Internet sites that have teased out enough of the details for us to work with.

The technical specifications I used were from:

- mikroElektronika **www.mikroe.com/forum/viewtopic.php?t=8792**
- *Nuts & Volts* September '03 column by Jon Williams
- Curious Inventor **http://store.curiousinventor.com/guides/PS2**

I will not repeat the communication protocol information available on these sites, except to discuss how it relates to our current project. If you want to dig deeper into the PlayStation controllers communication, you can use those references cited.

FIGURE 2. PS2 with wireless Playstation controller connected to six servos.

Construction

The PS2 controller has a nine-pin connector shown in **Figure 3**. If you are hardwiring this circuit from scratch, you need to obtain a PS2 extension cable. Leave about 4-5 inches of cable attached to the female connector also shown in **Figure 3**. Cut and strip the wires inside the cable to attach to your circuit.

Wire definitions are: 1-Brown is Data; 2-Orange is Command; 3-Grey is Vibration; 4-Black is Ground; 5-Red is Power; 6-Yellow is Attention; 7-Blue is Clock; 8-White is Unknown; and 9-Green is Acknowledge.

We use seven of the nine wires off the connector with the exceptions being the green and white wires. These are fine wires which are a little difficult to strip and solder. My fingers are a little clumsy for this detail work, so I prefer to use a pre-made PS2 interface cable with a seven-position socket that plugs onto a header. The schematic for our circuit is shown in **Figure 4**. You can hardwire and build this circuit on a prototyping breadboard (see **Figure 5**) or purchase the kit.

To build the kit, we start with the PCB (**Figure 6**). The kit is available in three versions that have different power supply ratings of either one ampere, three ampere, or five ampere. The higher amps can supply more power to the servomotors without overheating.

Begin construction by mounting components and parts on the silkscreen side of the PCB. R1 and R2 are 1K resistors (color bands are brown, black, red). R3 is a 4.7K resistor (color bands are yellow, purple, red); R4 is a 10K resistor (color bands are brown, black, orange). Check this out in **Figure 7**.

Next, mount and solder the ICS-18 microcontroller socket labeled U2. Then, mount and solder the 1N5402X diodes labeled D3–D7. Keep the line on the diode orientated to the line of the silkscreen outline of the diode. Next, mount the two colored LEDs. The red LED is labeled Power LED; the green LED is labeled Status LED. Orientate the LEDs properly. The longer lead of the LED is positive. This faces the round portion of the LED outline. The shorter lead of the LED is negative; it faces the flat side of the LED silkscreen outline as shown in **Figure 8**.

Now, mount and solder the round pushbutton switch

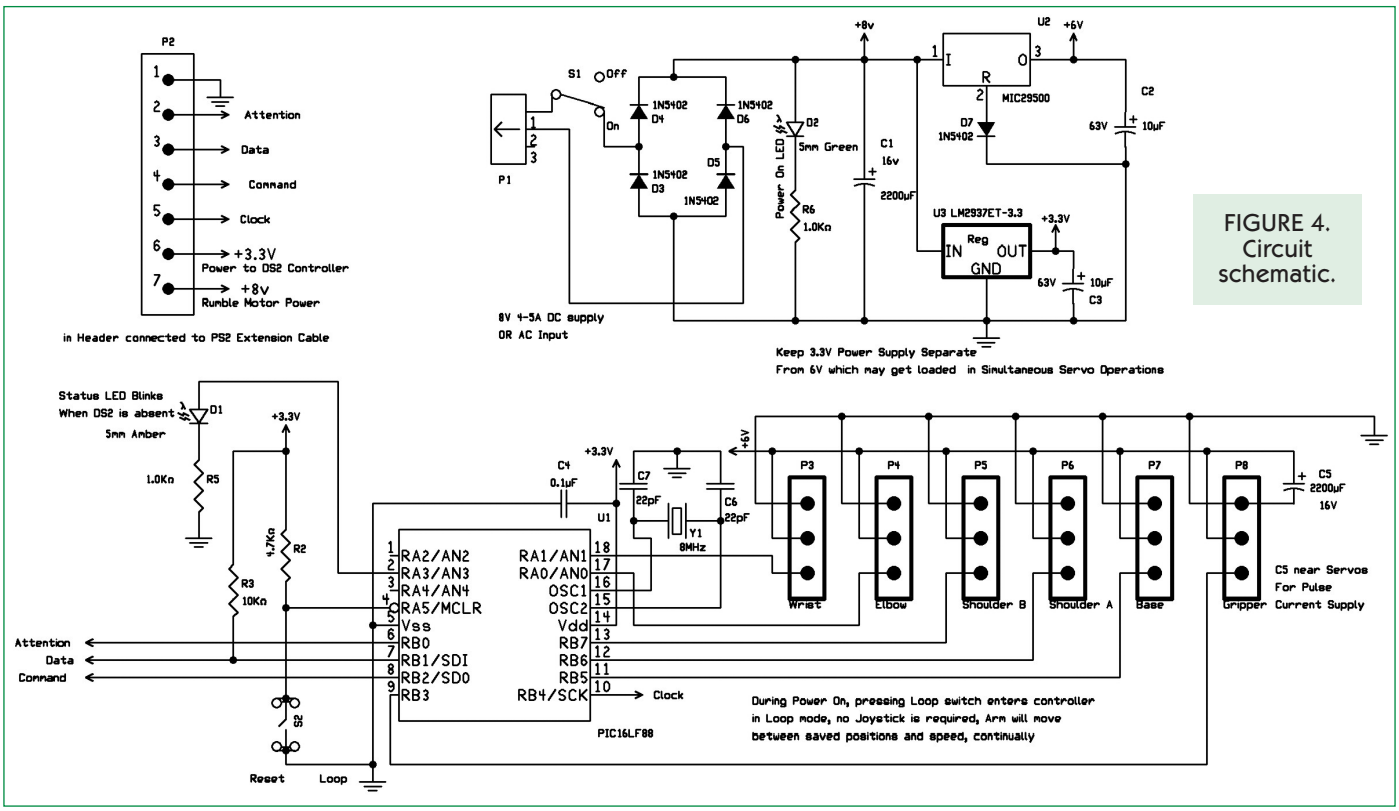


FIGURE 4. Circuit schematic.

***** Depending on which version of the PS2 controller you purchased, this regulator may be changed to the three amp or five amp version.

and the toggle switch labeled S1. Mount and solder the power supply jack labeled P1. When mounting capacitors, the longer lead is the positive lead. Capacitors C2 and C6 are 1,000 µf. C5 and C7 are 10 µf capacitors. The following capacitors are non-polarized and may go in with the leads facing either way. Capacitor C4 is a 0.1 µf 100V. C1 and C3 are small 22 pF capacitors. Next, mount and solder the 8 MHz crystal labeled Y1. Now, mount the SMH-03's labeled P2-P7, then mount the seven-pin header P8 and the LM2937 labeled U1. Before mounting the 7805* voltage regulator, first attach it to the heatsink (heatsink is included with the three and five amp versions only) and then mount it to the PCB in the position labeled U3 (see **Figure 9**).

When mounting voltage regulators LM2937 and 7805* to the PCB, the flat side must be aligned with the small rectangle printed on the board.

Plug the PS2 connector into the P8 connector. The black wire of the connector faces toward the right, with the PS2-SMC-06 facing you. Secure the PS2 connector to the top side of the PCB as in **Figure 10**.

The hex file for programming the 16F88 accompanies this article. If you purchased the kit, the PIC microcontroller is pre-programmed.

Power Supply

The PS2 SMC can use a variety of power supplies from 6V to 9V (AC/DC, one Amp, or more) thanks to the onboard rectifiers. You don't have to worry about DC supply polarity. On my prototype, power is supplied to the board through the 2.1 mm DC barrel jack. The circuit can be reset anytime using a momentary contact reset switch.

PS2 SMC Operation

Begin by plugging in nine-pin male connector of your PS2 controller or wireless receiver into the female connector on the circuit board.

Servomotor Connectors

There are six standard three-pin male headers for connecting hobby servomotors Servomotor 1 to Servomotor 6. Connect one to six servomotors as shown in **Figure 11**.

Power-Up

Plug in the PS2 controller, connect the servomotors, and toggle the on/off switch to the "on" position. The green LED will light indicating that power is on. The amber LED (status LED will go on after a couple of seconds) indicates a communication link has been established successfully with the PS2 and that the servomotors are now ready to be controlled. This amber LED will stay on. At the same time, the red LED on your PS2 will go on, indicating the PS2 is in analog mode. All six servomotors in normal Independent mode will move to the center neutral position on power-up.

When using the cordless PS2, make sure the communication link between the PS2 and its wireless receiver is established. Pressing any of the action buttons on the PS2 will help establish the wireless link on power-up. When a proper link between the PS2 and its wireless

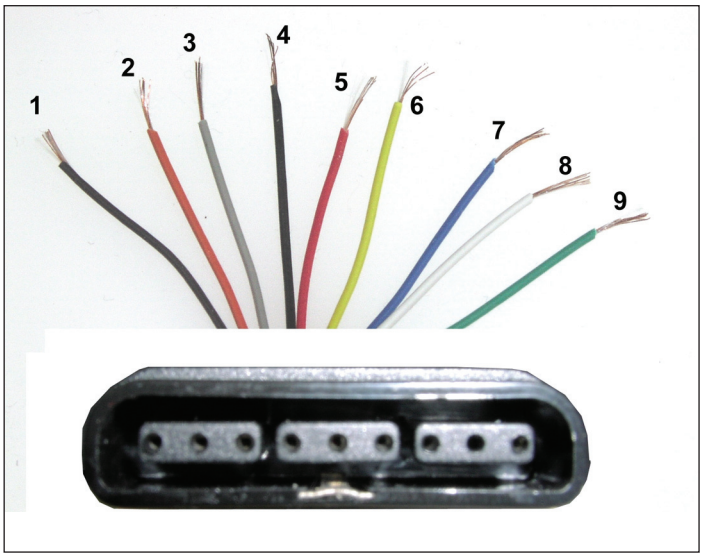


FIGURE 3. Playstation nine-pin connector with wires out.

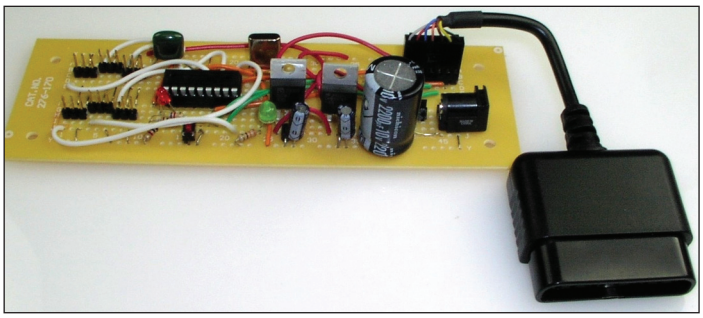


FIGURE 5. Prototype PS2 interface controller.

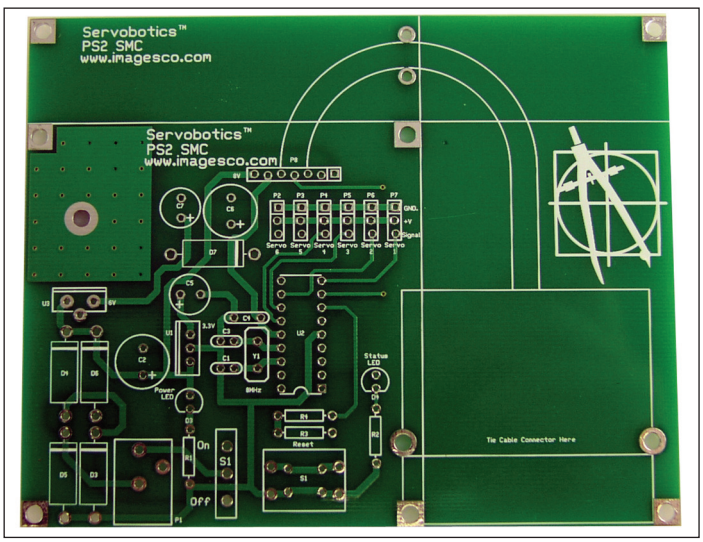


FIGURE 6. PCB for interface controller.

FIGURES 7-10. PCB construction.

FIGURE 7.

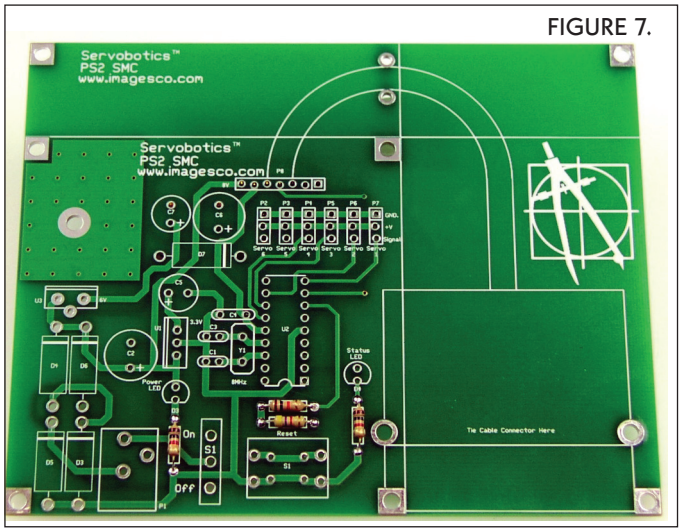


FIGURE 8.

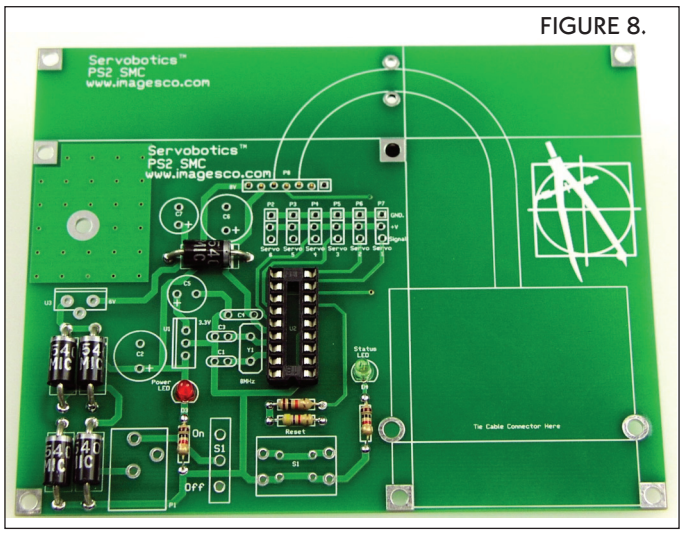


FIGURE 9.

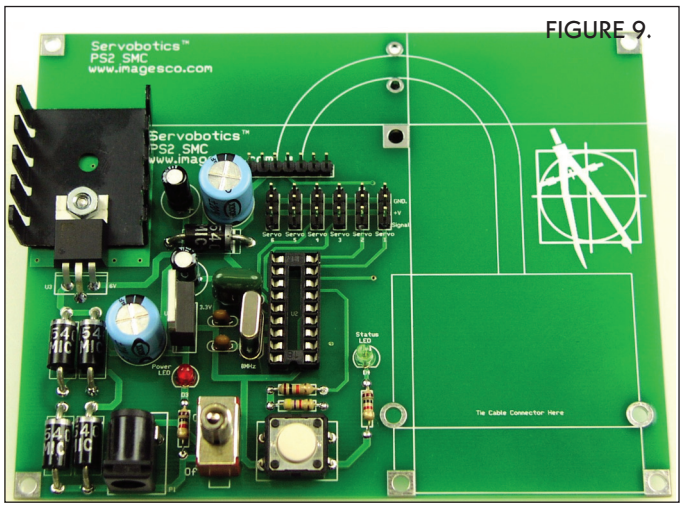
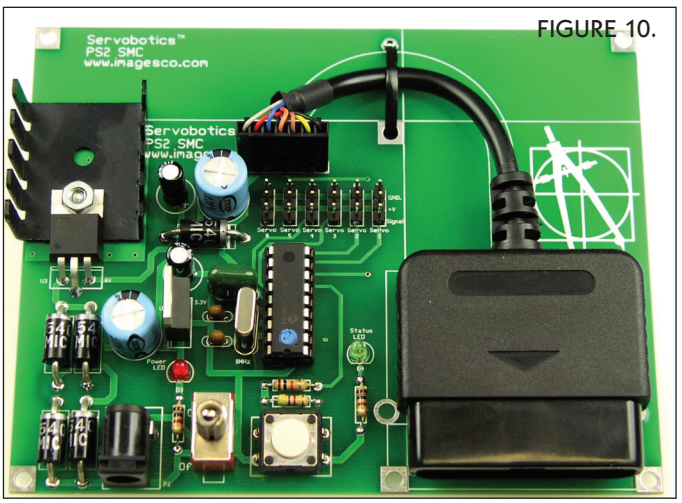


FIGURE 10.



receiver is established, pressing the action button will result in the LED blinking on the wireless receiver.

Controller Map

The six servomotors are controlled by the two analog joysticks (thumb sticks) on the PS2 controller; see **Figure 12**. Servomotor 1 and Servomotor 2 are controlled by the left analog joystick. The X axis of the left analog joystick controls Servomotor 1 and the Y axis controls Servomotor 2. Tilting the joystick along any axis will result in motion of the servomotor associated with that axis. Tilting the joystick in the opposite direction along the same axis will result in servomotor motion in the opposite direction.

Servomotor speed is proportional to the tilt angle of the joystick. Tilting the joystick slightly will result in the slowest servomotor motion. Tilting the joystick all the way will result in fastest servomotor motion. There are three levels of servomotor speed along one direction of the axis.

Releasing the joysticks will bring it to center position and the servo will stop at its current position.

Since sticks may not always return to the center neutral position, there is a dead band around the center of the stick which results in no movement. This ensures the servomotor doesn't keep moving in case the stick has not returned to the neutral position.

Servomotor 3 and Servomotor 4 are controlled by the right analog stick in the same manner as described for the left analog stick. Pressing Left 1 (shoulder button) in front of the PS2 will shift control of the right analog stick from

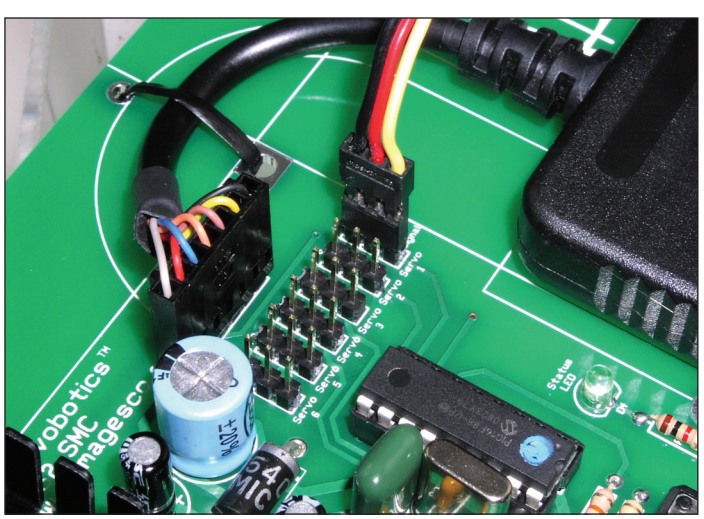


FIGURE 11. Servomotor connections.

Servomotor 3 – Servomotor 4 to Servomotor 5 – Servomotor 6. Servomotor 5 and 6 are controlled by the right analog stick as long as the left 1 button is pressed. Releasing the left 1 button will shift control of the right analog stick back to Servomotor 3 and 4.

Rumble

The servomotor has a maximum limit in both directions and driving it beyond these limits will cause damage to the servomotor and drain heavy current from the power supply. To avoid this, servomotors are not allowed to be driven beyond their Pulse Width (PW) limits, as determined by the manufacturer's recommendation for PW signals sent to the servomotor. When any of the servomotors have reached their limit and the user continues trying to move it by tilting the corresponding analog stick, the servomotor will not move and this condition will be indicated by a Rumble (force feedback) feature. Rumble will stay on as long as the user keeps the stick tilted.

Programmable Configuration Modes

The PS2 SMC can be configured in three different operating modes. These modes can be changed "on-the-fly" during normal operation. New mode selection is stored in the non-volatile memory of the PIC. The stored mode will automatically load the next time the PS2 SMC is turned on.

1.) Mobile Platform Mode – MPM (Circle and Square Buttons)

In this mode, Servomotor 1 and Servomotor 2 are configured as differential drive in a moving platform (for continuous rotation servomotors). To activate the MPM, press the Circle button.

Pressing the Square button will select the Normal Independent Mode (NIM) for servomotor control. This mode is for standard servomotors and not continuous rotation ones.

In MPM mode, the left analog stick will act as a navigation control with built-in speed control. Tilting the stick upwards will result in forward motion; downwards is for reverse motion; left is for left turn; and right is for right

FIGURE 12. Servomotors mapped to PlayStation controller.

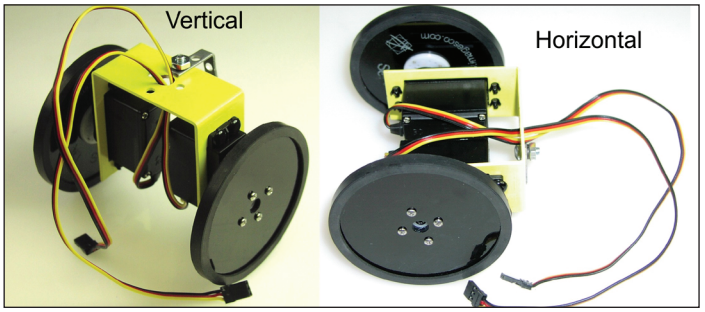


FIGURE 13. Mobile test platform.

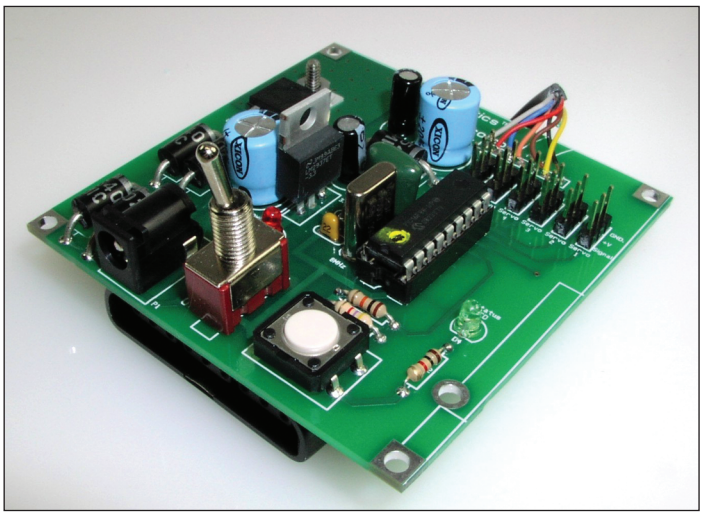


FIGURE 14. Reduced size PCB for mobile operations on robots.

Parts List

QTY	ITEM
(1)	Printed Circuit Board
(1)	ICS-18
(5)	1N5401-!N5402
(1)	Sub-min LED
(1)	Green LED
(6)	Three-Pin Straight Male Header
(1)	Seven-Pin Straight Male Header
(1)	8.0 MHz Crystal
(1)	7805 (Voltage Regulator)
(1)	HeatSink-03
(2)	1K ohm 1/4 watt Resistor
(2)	4.7K ohm 1/4 watt Resistor
(1)	10K ohm 1/4 watt Resistor
(1)	2.5 mm PC Power Jack Connector (PJ-102B)
(1)	12 mm Momentary Switch Tact (SW-25)
(1)	Toggle Switch SPDT On-On PCM (SW-07)
(2)	10 µF Capacitor, 16V
(1)	0.1 µF Capacitor, 100V
(2)	22 pF Capacitor, 50V
(2)	1,000 µF Capacitor, 10V
(1)	LM2937
(1)	PIC16F88 Microcontroller
(1)	Seven-Position Modular to CS Connector (Cable-PS2)
(1)	Jack-09
(1)	9 VDC 300 mA. (or greater) Power Supply

A complete or partial kit to go with this article can be purchased online from the SERVO Webstore www.servomagazine.com or call our order desk at 800-783-4624.

Technical Supplement PS2

PS2 Controller

Before going further, we need to first look into the details of what the SPI interface will be communicating to the host. This involves the PS2 control buttons and analog joysticks.

The PS2 has 16 buttons: four Directional Pad (Dpad) buttons; four Action buttons (circle, triangle, etc.); and four Shoulder buttons, (start, select etc.). Both analog joysticks can be pressed as buttons, in turn. When pressed, each button outputs -0; when released each button outputs -1.

The two analog joySticks on the PS2 can each be moved in two axis. The output of each stick is proportional to the amount of tilt the stick is placed under. Each stick outputs two values of 0 to 255, corresponding and proportional to the tilt on each axis.

Twelve of the PS2 button may be set as Pressure Sensitive buttons (four Dpad, four Action, four Shoulder), each outputting a value between 0 to 255, proportional to the amount of pressure exerted on the buttons.

We can mix and match the function of the buttons and sticks depending on which of the PS2 three modes of operation has been set up (either digital, analog, or analog pressure).

In digital mode, analog sticks don't output a value; pressure values are also not present. Only the on/off state of the digital buttons is outputted.

In analog mode, the digital on/off state is outputted, as well as the analog joystick values.

In analog pressure mode, digital on/off, analog sticks, and pressure values all are outputted.

The SPI to the host is a synchronous serial half/full duplex communication protocol. The PS2 controller is the slave and the host controller — in our case, the PIC microcontroller — is the master; and generates synchronizing clock. The master PIC sends command bytes to the PS2, and the PS2 sends data bytes back (full duplex mode).

Commands have a three-byte header. These bytes request the PS2 to acknowledge its current mode of operation. The PS2 responds with three data bytes indicating its current mode.

The current state of the buttons, stick, and pressure values can be requested from the PS2 by a polling method where a specific button value's command is sent and the PS2 responds with the current state.

Every command is divided into a sequence of bytes. The header is followed by data. The digital button's polling command, for example, is five bytes. So, to know button states, the master has to transmit five command bytes to the PS2; the PS2 will return five data bytes back to the master. In contrast, the analog polling commands consist of nine bytes.

In order to select the mode of operation, turn on/off the vibration motors the PS2 needs to be configured (with configuration commands).

PIC Program

The general function of the PIC program is as follows:

When power is applied, the PS2 default is the digital mode. The PIC program sends a wireless idle command to the PS2 that is specifically used for wireless. The command consists of five bytes and data returned is not used.

Commands to the controller should be separated by five to 10 msecs. However, there is no delay between bytes used in a command.

Next, the digital button's polling command of five bytes is sent to the PS2, for which the PS2 returns the button states in packets of five bytes. Here, the second data byte from the PS2 is a mode of operation byte. This byte should be $41 hex. If it isn't, either the PS2 is absent or is faulty. So the mode byte from the PS2 is used as a reference and benchmark for proper communication. This command is only to make sure the PS2 is up and working.

Next, the PS2 should be configured for the proper mode of operation. For that, the configuration enter command is given to the PS2.

Once into the configuration setup, subsequent commands such as Vibration Motors On, Pressure Values On, and Analog Lock Mode are given to the PS2 sequentially. Once the configuration is set up to our requirements, the exit command is given to the PS2.

Here are a few things you should know:

1. In the configuration mode, if the PS2 is not sent a command for some time, it automatically falls out of configuration mode.
2. The configuration enter command length is equal to that of the current mode of operation. For example, if the current mode is digital, the configuration enter command will have the same number of bytes as that of the digital polling command.
3. Once into configuration mode, all configuration commands (including configuration exit commands) are nine bytes in length.
4. The PS2 will output the current mode in a data string (second byte) in response to a configuration enter command. After that, it will output configuration mode bytes in data strings.

After configuration, a command is again set to the required length for the analog - pressure poll command i.e., 21 bytes and various command bytes are initialized as per the requirements. Next, comes the servomotor control loop. Every 20 msecs, the PS2 is polled by the analog pressure poll command, and a new servomotor value is calculated based on current button/stick states; new values are pulsed out.

We don't use pressure-sensitive buttons in our program.

For each command, every byte has to be per the specifications. You will notice that for each command, command bytes are initialized, the byte length of a command is set, and then the SPI subroutine will be called. The SPI subroutine will return data bytes.

If after configuration, the PS2 doesn't function, it has to be reconfigured again. So, in the servo loop, the second data byte (mode byte) is checked every time the PS2 is polled. If that byte is not what we expect it to be, the PS2 is reconfigured again. The PS2 also needs to be polled for a minimum number of times in one second period. Failing to do so will result in the PS2 going back into the default state.

We are able to maintain a 125 kHz clock frequency for the synchronous clock in SPI between the PIC and PS2.

General Notes

1. One should plug the PS2 controller and servomotors on the interface board before powering up. Only remove them after turning off the board.
2. Configuration modes can be changed anytime the user wants in normal operation. However, since the associated mechanical hardware connected to the motors for different modes is so different, practically speaking, a need for changing from one mode to another won't arise during an operation. Mode change can be done when required immediately at power-on, once the status LED goes on. This will ensure that the servomotors are at the center position when the mode is changed so an unwanted movement of the servomotors won't occur as a result of mode change.
3. Servomotor control is inhibited when the mode change buttons are pressed.

turn of your moving platform. The tilt angle of the stick will determine the speed. Two-level speed control is implemented.

Sometimes the stick doesn't return to the neutral position on release which may result in unwanted movements of the servomotors. Creating a small dead space or band around the center position of the joystick eliminates this movement. Within this dead band, the servomotors aren't activated. To test this mode, I created a quick and dirty mobile platform using two continuous rotation (CR) servomotors and a dual servomotor bracket (see **Figure 13**).

For mobile operations or if you're just tight on space, the PCB can be cut down (before assembly) so its footprint is much smaller (**Figure 14**). The PS2 connector is secured

to the bottom of the PCB.

2. Opposite Phase Mode – OPM (Cross and Triangle Buttons)

Servomotor 3 and Servomotor 4 can be configured to be controlled independently or in Opposite Phase Mode (OPM). In OPM, each servomotor shaft moves in the opposite direction from the other, 180 degrees out of phase. Pressing the Cross button will select OPM mode. Pressing the Triangle button will select Normal Independent Mode (NIM).

OPM can be helpful when two servomotors are connected to a common load, or when you're using two servomotors for increasing the overall relative speed of movement.

In OPM, the X axis of the right analog stick will control Servomotor 3 and 4. Tilting the stick along the X axis in one direction will result in the simultaneous opposite movement of Servomotor 3 and Servomotor 4 at the same speed.

Tilting the stick in the opposite direction will result in the simultaneous opposite movement of Servomotor 3 and 4, but in the opposite direction to that of the previous case. As in MPM, there is a dead band around the center of the stick and three-level speed control proportional to the stick tilt.

Servomotor 5 and Servomotor 6 always operate in NIM.

3.) Normal Independent Mode – NIM (Square and Circle Buttons)

Pressing the Square button will select NIM for servomotor control. This mode is for standard servomotors and not CR servomotors.

Handiness (Righties and Lefties)

When selected, MPM mode by default sets the left analog stick to control Navigation. For righties, they can switch Navigation control to the right analog stick by pressing the right arrow button on the directional pad. Pressing the left arrow button on the directional pad brings the stick controls back to their default mapping. In Righty mode, the left analog stick will control Servomotor 3 and Servomotor 4; the right stick controls Servomotor 5 and Servomotor 6.

When the Shoulder button is pressed,the right analog stick will control Servomotor 1 and Servomotor 2. Basically, Leftie and Righty mode are independent of other modes selected; control between the two sticks is swapped, irrespective of the mode.

Wrap-Up Mode

Utilizing existing equipment in your robot projects is always satisfying. Using the PS2 controller and joysticks adds to the enjoyment of deploying your builds. SV