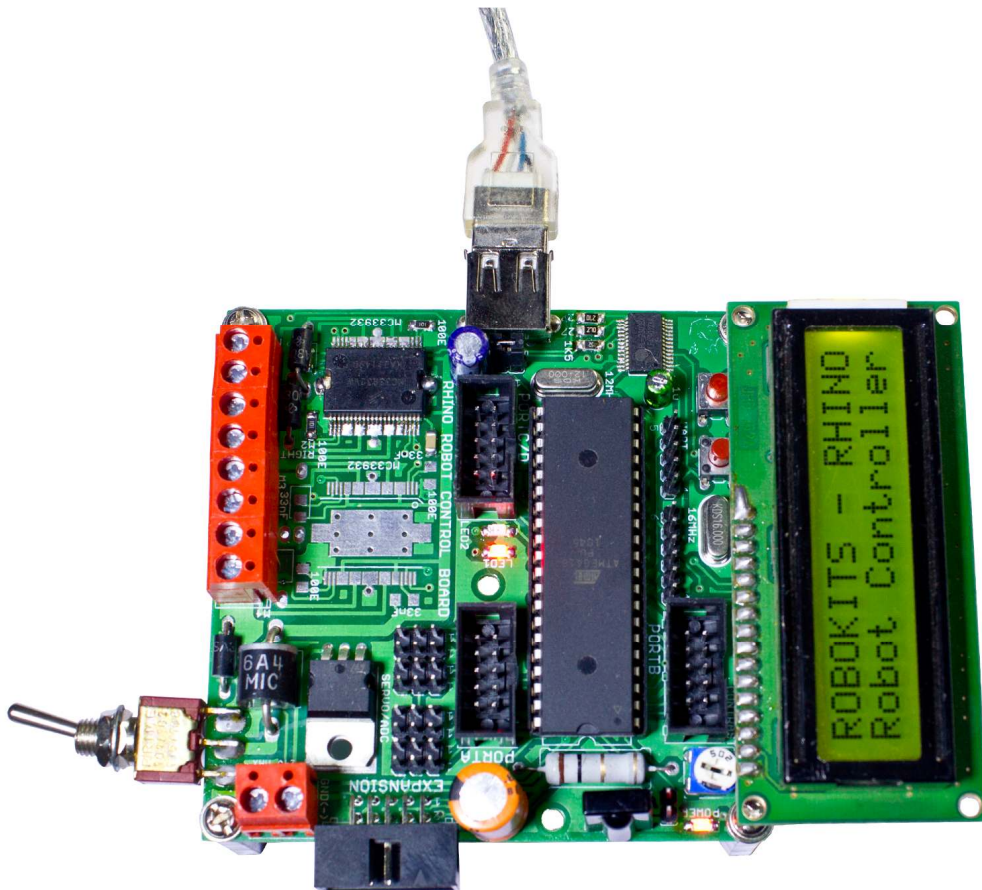


Rhino Robot Control Board

PS2 Remote Controlled Robot



RAN1104: Building a Manual Playstation Remote Controlled Robot

Robokits India

<http://www.robokits.co.in>
info@robokits.co.in

Rhino Robot Control Board is our most powerful, versatile and most easy to use robot control board. In this first part of tutorials and application notes series you will learn about basic elements of Rhino Board and Quick C – IDE software which is specially developed software for this board.

We also hereby assume that you have gone through [Rhino Board Manual](#), [Quick C IDE user manual](#) and [Quick C IDE library reference](#). It's not necessary for you to understand everything written in those documents but you should have an overview so that you can use them as reference for some part in this document.

PlayStation controller is actually a Video Game controller remote for Sony's PlayStation 2 and PlayStation 3 gaming consoles. The wireless variant of this controller is used widely for manual robot control. There are many reasons for this, first is its cheap and easily available, second is it has 16 buttons and 4 analog inputs which means a great control over robots and connected accessories. We will be using the same remote for our application. The bad part is these remote controls don't come with easily usable pin outs. They have a connector for PlayStation compatible connector. So you will either need to buy a readymade modified controller or you will need to buy a controller and modify it yourself. Also note that Wired PlayStation gamepad can also be used with this board provided that all connections are correct.

This tutorial covers

- **Accessing all buttons and Joysticks of Remote Controller**
- **Making a Wireless PlayStation Remote Controlled Robot**
- **Making a Wireless PlayStation Remote Controlled Robot with Pan-Tilt and preprogrammed functions**

Required Items

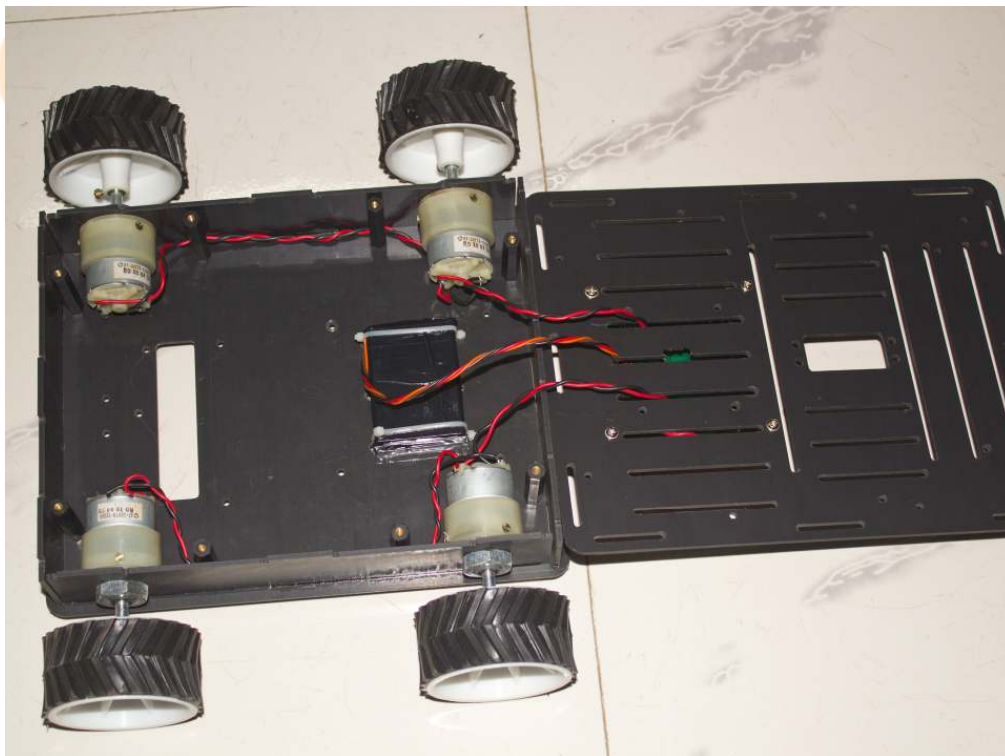
REQUIRED ITEM	SUGGESTED ITEM/USED IN THIS TUTORIAL
Rhino Board	<u>Rhino Robot Control Board - AVR Based with Quick C Compiler</u>
Robot Chassis	<u>Open Source Multipurpose Robot Platform Chassis kit</u>
Motors	<u>300RPM 12V DC Motor with Gearbox</u>
Battery	<u>Lithium-Ion Rechargeable Battery Pack 11.1V 1500mAh</u>
Wheels	<u>Tracked wheel for DC motors 4cm width</u>
Modified PlayStation Wireless Remote	<u>Playstation 2 robot controller</u>
Optional : 2 Servo Pan-Tilt	<u>2 Axis Pan tilt for Sensors and wireless camera</u>

*We highly recommend above parts as the application note is specifically build using all of above items. It's not necessary that you use the same items listed above. You can also use any similar/higher configuration items. However you may need to make some changes to the code to make necessary adjustments.

Items used for in this tutorial



General purpose robot chassis with 4 300 RPM motors and 4 cm width wheels



Insied view



With Pan-Tilt and PS2 Wireless Receiver



PS2 Wireless Controller (Modified)



We have chosen a general purpose chassis with 4 motors. An 11.1V 1500mAh battery is mounted inside with cable ties. Rhino Board is mounted on top and Battery and motor connections are made to the board. A modified PlayStation Wireless Remote is used.

PlayStation Transmitter

There are in total 16 accessible buttons on transmitter. 4 is digital keypad on left side which is called UP, DOWN, LEFT and RIGHT. Other 4 buttons are on right side which are named A, X, D and O. 4 buttons are on top called L1 and L2 on left side and R1 and R2 on right side. There are 2 buttons in center for Start and Select and 2 more when you press joystick which are called JL and JR.

Apart from these there are 2 Joysticks which are 2 axis ones. So both joysticks will give total of 4 analog outputs which ranges from 0 to 255. When they are in normal position (center position) they give value 127 or 128.

PlayStation Receiver

The wireless PlayStation remote has a receiver which will be connected to Rhino board. The connection should be done on the LCD connector. Its obvious that LCD will not work if PSx receiver is connected. PS2 Receiver used here has following pinouts.

Pin Number	Wire Colour	Pin Description
1	Black	Ground for power supply and signal
2	VCC	+5V input
3	Red	Clock
4	Orange	Attention
5	Yellow	Command
6	Green	Data

Detailed working of protocol is out of the scope of this document but you can search on the internet for the same.

The main thing to understand is the receiver is continuously communicating with Rhino Board and the transmitter. Rhino Board should continuously send commands for polling data from receiver. This is done through function `PSXPOLL()`; . This function updates input data to all PSx variables like `LJOYX`, `LJOYY`, `RJOYX`, `RJOYY`, `PSX_L1`, `PSX_ST` etc.

CODE

Code 1 : Accessing PSx Joysticks and Buttons

Open **16 - PSx to UART Demo** program in sample codes folder.

Library used : UART, PSX

The code is simple for accessing PSX input and displays it on UART. You can use a terminal software to see data coming on UART. You can download the software here :

http://robokits.co.in/downloads/Robokits_Super_Terminal_Setup_without_dotnet.exe

Once you set parameters (COM Port to your Rhino Port, see while programming and baudrate to 115200) and click connect data will start coming to the screen through UART. If your PC is getting slow while showing data you can disable HEX and DECIMAL view to save resources in Super Terminal.

While you are getting values continuously on your screen you may press different buttons and move joysticks to see the inputs to Rhino.

Make sure you click Disconnect after testing because until the port is open you will not be able to program Rhino.

Code 2 : Making a Wireless PlayStation Remote Controlled Robot with or without Pan Tilt

Open **017 - PSx controlled robot with pan tilt** program in sample codes folder.

Library used : PSX, Servo

Using the above code this code uses the input from PSx to control motors and servos. Speeds of motors are controlled from input from analog joysticks. There are also some formulas written for calculations. This ensures that motor will get minimum PWM to make the robot moving. Generally at low PWM motors don't start to avoid this formulas are written.

You can also see that some functions like **SERVO[0]**, **SERVOSPEED[]**, **SERVOMAX[]**, **SERVOMIN[]** are used. These are for pan tilt servos. There are 2 servos used in this example one for Panning and one for Tilting. Pan servo is controlled through D and O buttons while Tilt servo is controlled by A and X button. You can also see that we have also utilized start button to move both servos to center position.

This code can also be used without servos for normal remote controlled robot.



Service and Support

Service and support for this product are available from Robokits India. The Robokits Web site (<http://www.robokits.co.in>) maintains current contact information for all Robokits products.

Limitations and Warrantees

The **Rhino Robot Control Board** is intended for personal experimental and amusement use and in no case should be used where the health or safety of persons may depend on its proper operation. Robokits provides no warrantee of suitability or performance for any purpose for the product. Use of the product software and or hardware is with the understanding that any outcome whatsoever is at the users own risk. Robokits sole guarantee is that the software and hardware perform in compliance with this document at the time it was shipped to the best of our ability given reasonable care in manufacture and testing. All products are tested for their best performance before shipping, and no warranty or guarantee is provided on any of them. Of course the support is available on all of them for no cost.

Disclaimer

Copyright © Robokits India, 2012

Neither the whole nor any part of the information contained in, or the product described in this manual, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder.

This product and its documentation are supplied on an as-is basis and no warranty as to their suitability for any particular purpose is either made or implied.

This document provides preliminary information that may be subject to change without notice.