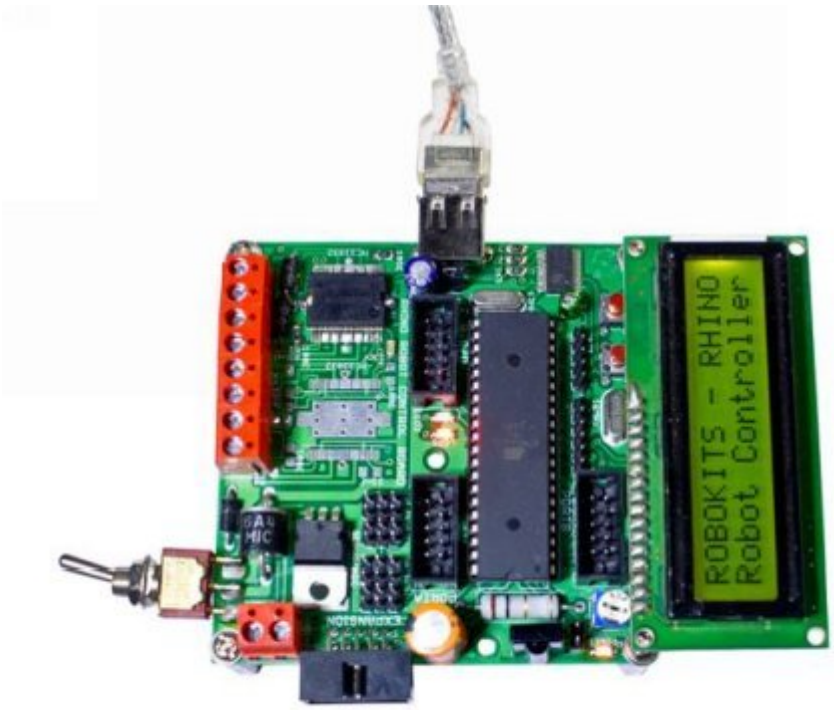


RAN1103: IR Wireless Remote Controlled Robot

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. This tutorial series is intended for giving the idea of versatility and use of this board in many types of robots.

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.

All files required for this tutorial could be downloaded from here : http://robokits.co.in/documentation/RAN-1103.zip .

This tutorial covers

- **Making a Wireless IR Remote Controlled Robot**

Required Items

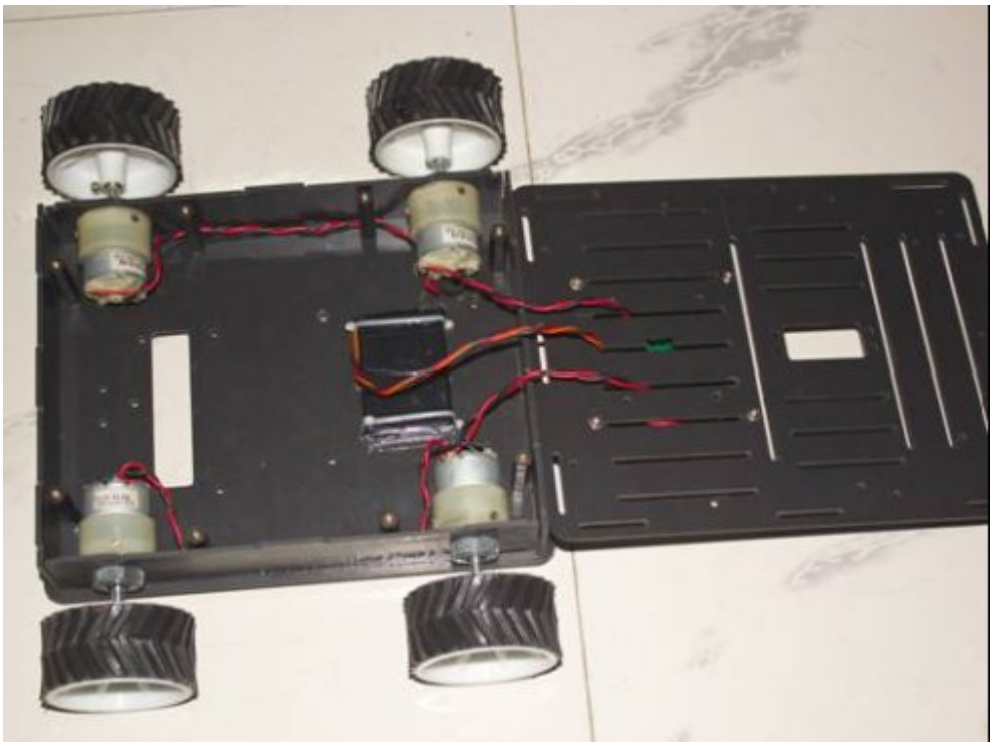
REQUIRED ITEM	SUGGESTED ITEM/USED IN THIS TUTORIAL
Rhino Board	http://robokits.co.in/shop/index.php?main_page=product_info&products_id=312

Robot Chassis	http://robokits.co.in/shop/index.php?main_page=product_info&products_id=378
Motors	http://robokits.co.in/shop/index.php?main_page=product_info&products_id=50
Battery	http://robokits.co.in/shop/index.php?main_page=product_info&products_id=69
Wheels	http://robokits.co.in/shop/index.php?main_page=product_info&products_id=297
IR Remote	http://robokits.co.in/shop/index.php?main_page=product_info&products_id=168

*** It's not necessary to use the same items, you can use any similar items. However some coding may need to be changes as per hardware if it's different this listed above.**

Items used for in this tutorial

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



Inside view

IR Remote

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 TV remote control which is working on RC5 protocol is used.

Which remote to choose? I have a remote, How do I test is it compatible?

Any remote with RC5 protocol will work with this board. Most TV remotes work on this protocol. If you are not sure what codes each button is transmitting check code 013 - IR Remote Tester in sample codes folder. GETRC5() function will return received RC5 code on call. If your remote is compatible with Rhino board it will show some value less than 255. If you get different values on different buttons that remote will work.

This code will also help to know codes for each button. Different remotes transmit different codes on same buttons, so you will need to know first which buttons you will be using to control the robot and what are the codes for that.

If you don't get any number on LCD display after flashing this code, your remote will not work.

Key codes for specific buttons

The remote which is shown above in picture, we chose CH+ button to move forward, CH- Button to move backward, VOL+ button to turn right and VOL- Button to turn left. Use buttons 1, 2 and 3 to change speeds.

When this remote was tested with remote tester code it returned these values

Button	RC5 Code
CH+	32
CH-	33
VOL+	16
VOL-	17
1	1
2	2
3	3

CODE

Code 1 : IR remote controlled robot (without speed control)

Code is similar to previous tutorials manual robot control code. Only difference is the input. Instead taking input from switches we will be taking inputs from a Remote control which is already tested and codes are known for each switches we will be using.

You can open 014 - IR Remote Controlled Robot code in sample codes folder.

Library used in this code : IO Notations, Motor – Motor 1 & 2 Active with PWM

Here num=GETRC5(); statement takes RC5 input and stores in num variable. Then in the same loop the if-else if tree checks for proper key press and action to be done upon pressing it. Here a variable delay is used which is set to 85ms. The GETRC5() function needs around 80 ms to process so till that time motors should remain on otherwise we get jerks in driving motors.

This code has fixed speed of 100 so motors will run continuously on full speed. If we need speed control over the same, see next code.

Code 2 : IR remote controlled robot (with speed control)

Open 015 - IR Remote Controlled Robot with Speed Control in sample codes folder.

Library used in this code : IO Notations, Motor – Motor 1 & 2 Active with PWM

Here the whole code is same the only difference is the speed is variable and changed through keys 1, 2 and 3. 1 is minimum and 3 is maximum speed.

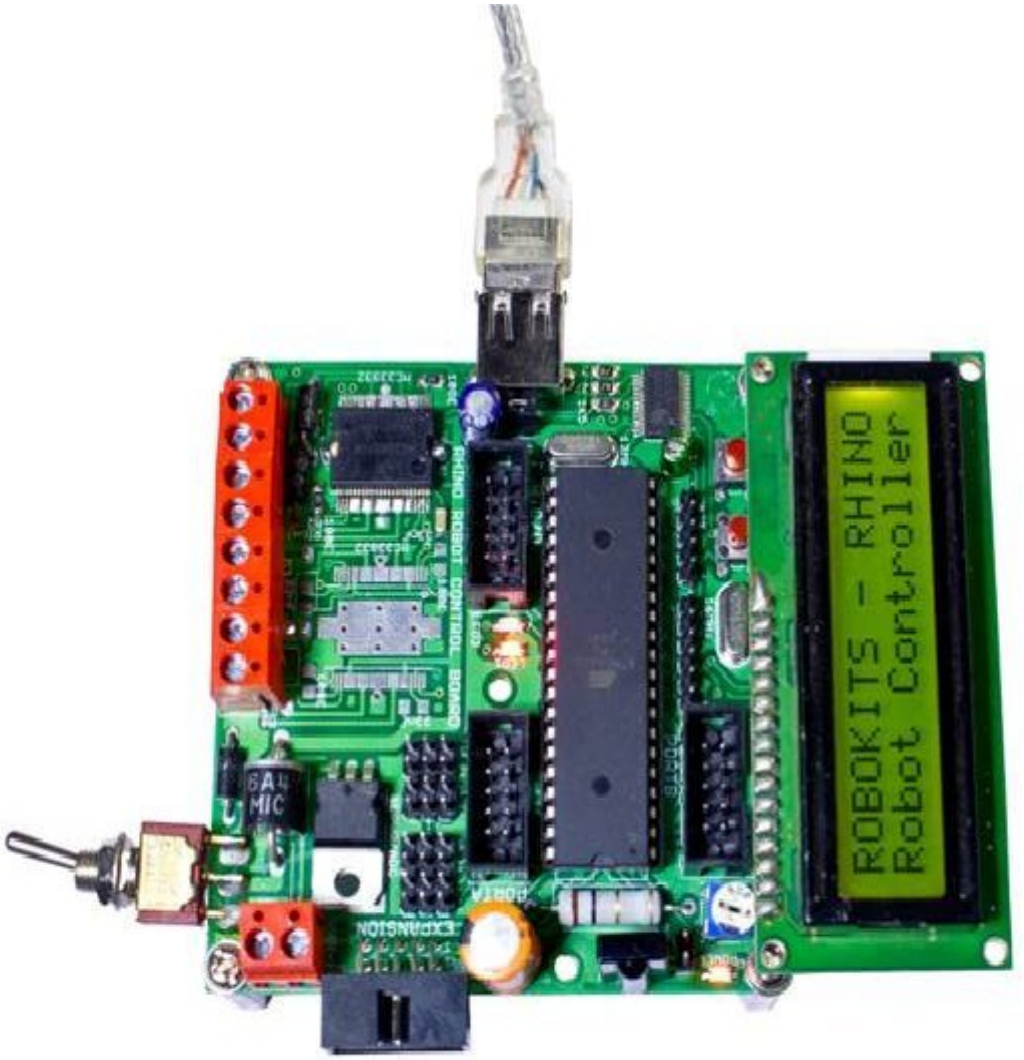
\

Rhino Robot Control Board Basics Tutorial

Table of Contents

Rhino Robot Control Board

Basic Tutorial



RAN1101 Basics : LEDs, Switches, LCD, Motor Drivers, UART

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

We also hereby assume that you have gone through Rhino Board Manual, Quick C IDE user manual and Quick necessary for you to understand everything written in those documents but you should have an overview so for some part in this document.

This tutorial covers

- **Controlling LEDs and give outputs**
- **Taking Inputs from switches**
- **Display static and dynamic data on LCD**

- Control motors
- Input and output data to UART (PC as other device)

Required Items for this tutorial

Rhino Robot Controller Board	http://robokits.co.in/shop/index.php?main_page=prod http://robokitsworld.com/index.php?main_page=prod
or Rhino Robot Controller Board L293	http://robokits.co.in/shop/index.php?main_page=prod http://robokitsworld.com/index.php?main_page=prod
LCD Display	http://robokits.co.in/shop/index.php?main_page=prod http://robokitsworld.com/index.php?main_page=prod

Downloads

RAN1101 - Rhino Board Basic features

1. LEDs

There are 2 LEDs on Rhino Board. LEDs are connected to PC6(IO 22) and PC7(IO 23) of Atmega16 MCU on board. There are functions in

Code 1 : Simple blink

Open 001 – LED1 blink program in sample codes folder.

Library used : Delay

The code is simple. LED1ON() turns on LED1 and LED1OFF() turn off LED1. DELAYMS(100) provides de

Code 2 : Simple blink with TOGGLELED function

Open 002 – LED Toggle program in sample codes folder.

Library used : Delay

The TOGGLELED1() turns on LED1 if its OFF, but if its On it will switch it off. This program creates same effect as code 1.

Code 3 : Control LEDs with PORT IO function

Open 003 – LED control with PORT IO program in sample codes folder.

Library used : Delay, IO Notations

This code uses PINMODE(23,1) and DIGITALWRITE(23,1) which are Arduino like pin access functions

The PINMODE(23,1) function sets PORTC.6 pin as output pin. DIGITALWRITE(23,1) Turns LED1 on As LED is connected to this pin it turns on and DIGITALWRITE(23,0) tunrs off as per pin status. Any other pin on board can be set high or low with this function.

This concludes the LED sections. It also shows how to set output on any pin of Rhino Board.

2. Switches

There is one general purpose switch on rhino board which is connected to PD.6 (IO 30). More switches may be connected to extra IOs and can be used by PORT IO functions. You may also use 8 switch Keypad or 4x4 Keypad with this board to get more switches. Second switch on Rhino is a reset switch which resets microcontroller.

Code 1 : Simple Input

Open 004 – Simple Switch Input program in sample codes folder.

Library used : Delay

Here to take input we need to use a conditional statement like 'if'. Here if(SWITCH1ON()) returns true(1) if switch is not pressed. The if – else condition creates logic to turn on and off the LED as per switch input.

This program will also work without DELAYUS(100) statement but its always advisable to keep delay in infinite loop to avoid microcontroller to use its all resources.

Code 2 : Input with PORT IO functions

Open 005 – Switch Input with PORT IO program in sample codes folder.

Library used : Delay, IO Notations

This code uses PINMODE(30,0) and DIGITALREAD(30) which are Arduino like pin access functions for Rhino Board. Y
reference file for all IO pins. Switch is connected on PORTD-6.

The PINMODE(30,0) function sets PORTD.6 pin as input pin. if(DIGITALREAD(30)) returns true(1) if switch is not pressed and false(0) if switch is pressed. This program runs same as previous one but using this functions you can connect switches externally to any of IOs and take input.

This concludes the Switch Input section. It also shows how to take digital input from any IO pin of Rhino Board.

3. LCD & ADC

Rhino Board is capable of driving a Character LCD display with parallel interface. Display size can be 16X2, 16X4, 20X2, 20X4 and other
compatible displays. For other types of LCDs like graphic LCD readymade functions and pinouts are not available, however it can be done easily like its done on any other AVR board.

Connection is very simple, Rhino board haves a 16 Pin Male header on one of the edges. Here an LCD display with 16 female header can be plugged in. There is a potentiometer for contrast adjustment.

Code 2 : Advanced LCD functions and displaying variables on LCD

Open 007 – LCD Advance program in sample codes folder.

Library used : LCD, Delay

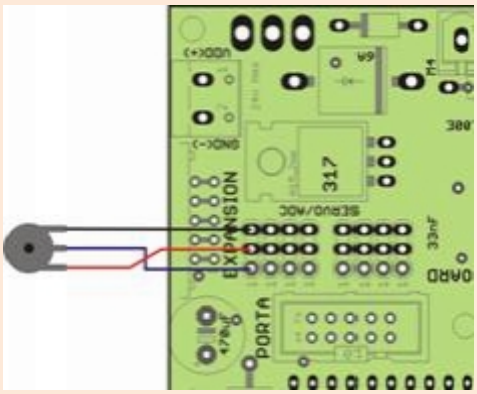
LCD is often used as debug tool while making programs. This may be to verify that whether ADC(Analog to digital convertor) is taking correct values from sensors or to verify formulas for calculations. This code prints tables of 1 to 99 on LCD screen line by line. This code also demonstrates the use of for loop and nested loops. LCD_PRINT function can't display values directly on LCD so first we need to convert any value to string and then pass through function. In this code itoa function is used to covert integer values to string.

Code 3 : Use Analog to digital convertor to take analog input and show input on LCD

Open 008 – ADC on LCD program in sample codes folder.

Library used : LCD, Delay, ADC

Here we will use a potentiometer to give variable voltage output from 0-5V and take that input, covert it to



You can also test this code without potentiometer, however it will show some random values. If you short the ACD pin to +5V or GND it will show 1024 and 0 values.

When above connection is done potentiometer will output 0-5V as shaft is rotated. This output goes to ADC0 Pin(see diagram). num=GETADC(7); takes the ADC input and stores to num variable. Next statements show it on LCD. This is infinite loop as used.

In place of potentiometer any sensor or equipment which gives analog output in 0-5V range can be connected to this board.

This concludes the LCD and ADC sections.

4. Motor drivers

Rhino Board includes motor drivers either L293 or MC33932. L293 provides 1A per motor while MC33932 provides 5A per motor
coding remains same for both boards, so program written for one board will work on other too.

For driving motors you will also need external power supply. All above sample codes will work without extra power supply on USB power motors will need external power.

Code 1 : Driving DC motor with direction and speed control

Open 009 – DC Motor Demo program in sample codes folder.

Library used : LCD, Delay, Motor – M1 & M2 Motors Active with PWM

This code controls DC motor with speed and direction control. Motor is connected to M1 Connector. LMF(i) of i and also LCD statements show current speed on LCD.

When program is running motor will start moving at speed of 30 and go till 99, once reached at speed 99 it will stay there for 2 seconds and then it will start reducing speed till 30, after speed 30 is command. After achived. Motor will stop by LMS()
in backward direction.

Code 2 : Driving Stepper motor with direction and speed control

Open 010 – Stepper Motor Demo program in sample codes folder.

Library used : LCD, Delay, Steppr – 1 Stepper Motor

Connection : Coil 1 of stepper is connected to M1, Coil 2 is connected to M2. Motor is used in bipolar mode so only 4 wires are used. This code controls Stepper motor with speed and direction control. Works just like previous code.

There are many more functions for motor control which are described in Library reference document.

5. UART

UART is most simple and most used communication protocol for microcontrollers. Rhino uses the same to communicate with PC. Also other devices can be connected on UART lines. UART lines are already connected to onboard USB-Serial convertor IC PL2303 through which Board connects to a USB port of PC. Not only this, UART protocol also allows PC to program Rhino Board.

Code 1 : UART – Input and Output data

Open 011 – UART Demo program in sample codes folder.

Library used : Delay, UART

The code uses inbuilt UART functions to communicate with serial devices. On PC you can use a terminal software and connect it to virtual
com port generated by Rhino Board. Once connected pressing reset switch will show a 2 line message on screen. After that sending any character on port will echoed back by board.

Check the first line where there is a definition for baud rate. This is the baud rate for communication. Make

Service and Support

Service and support for this product are available from Robokits India. The Robokits Web site (http://www.robokits.co.in 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 persons may depend on its proper operation. Robokits provides no warrantee of suitability or performance for any product software and or hardware is with the understanding that any outcome whatsoever is at the users own risk. software and hardware perform in compliance with this document at the time it was shipped to the best of our ability
and testing. All products are tested for their best performance before shipping, and no warranty or guarantee is