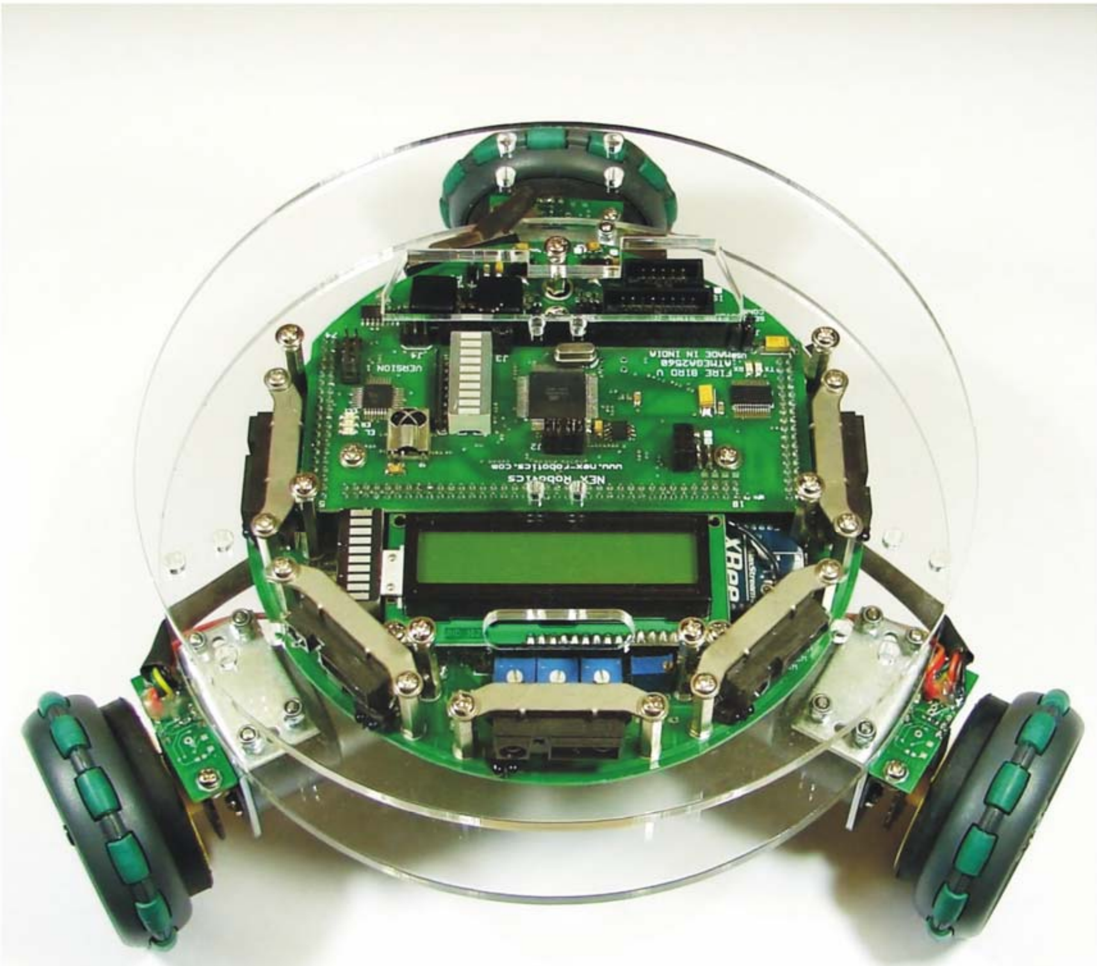
FIRE BIRD V

ATMEGA2560 OMNIDIRECTIONAL ROBOTIC RESEARCH PLATFORM USER GUIDE



Designed By:

ERTS Lab, CSE, IIT Bombay
www.it.iitb.ac.in/~erts

NEX ROBOTICS
www.nex-robotics.com

Manufactured By: NEX Robotics Pvt. Ltd.

FIRE BIRD V

OMNIDIRECTIONAL ROBOT
USER GUIDE

Version 2.00
December 3, 2010

Documentation author
Sachitanand Malewar, NEX Robotics Pvt. Ltd.
Anant Malewar, NEX Robotics Pvt. Ltd. and M. Tech, IIT Bombay

Credits (Alphabetically)
Aditya Sharma, NEX Robotics
Amey Apte, NEX Robotics
Anant Malewar, EE, M.Tech, IIT Bombay
Ashish Gudhe, CSE, M.Tech, IIT Bombay
Behlul Sutarwala, NEX Robotics
Gaurav Lohar, NEX Robotics
Gurulingesh R. CSE, M.Tech, IIT Bombay
Inderpreet Arora, EE, M.Tech, IIT Bombay
Prof. Kavi Arya, CSE, IIT Bombay
Prof. Krithi Ramamritham, CSE, IIT Bombay
Kunal Joshi, NEX Robotics
Nandan Salunke, RA, CSE, IIT Bombay
Pratim Patil, NEX Robotics
Preeti Malik, RA, CSE, IIT Bombay
Prakhar Goyal, CSE, M.Tech, IIT Bombay
Raviraj Bhatane, RA, CSE, IIT Bombay
Rohit Chauhan, NEX Robotics
Rajanikant Sawant, NEX Robotics
Saurabh Bengali, RA, CSE, IIT Bombay
Vaibhav Daghe, RA, CSE, IIT Bombay
Vibhooti Verma, CSE, M.Tech, IIT Bombay
Vinod Desai, NEX Robotics

Notice

The contents of this manual are subject to change without notice. All efforts have been made to ensure the accuracy of contents in this manual. However, should any errors be detected, NEX Robotics welcomes your corrections. You can send us your queries / suggestions at info@nex-robotics.com

⚠ **Robot's electronics is static sensitive. Use robot in static free environment.**

⚠ **Read the hardware and software manual completely before start using this robot**

Recycling:

Almost all of the robot parts are recyclable. Please send the robot parts to the recycling plant after its operational life. By recycling we can contribute to cleaner and healthier environment for the future generations.

Important:

1. Use this Application note with the Fire Bird V Hardware and Software Manual.

2. User must go through the Fire Bird V's Hardware and Software manuals before using the robot.

3. Crystal of the ATMEGA2560 microcontroller is upgraded to 14.7456MHz from 11.0592Mhz in all the Fire Bird V ATMEGA2560 robots delivered on or after 1st December 2010. This documentation is made considering crystal frequency as 14.7456MHz.

1. Introduction

Thanks for choosing the Fire Bird V mobile robot platform. Fire Bird V will give you good exposure to the world of robotics and embedded systems. Thanks to its innovative architecture and adoption of the 'Open Source Philosophy' in its software and hardware design, you will be able to create and contribute to complex applications that run on this platform, helping you acquire expertise as you spend more time with them.

Safety precautions:

- Robot's electronics is static sensitive. Use robot in static free environment.
- Read the assembling and operating instructions before working with the robot.
- If robot's battery low buzzer starts beeping, immediately charge the batteries.
- To prevent fire hazard, do not expose the equipment to rain or moisture.
- Refrain from dismantling the unit or any of its accessories once robot is assembled.
- Charge the NiMH battery only with the charger provided with the robot.
- Never allow NiMH battery to deep discharge.
- Mount all the components with correct polarity.
- Keep wheels away from long hair or fur.
- Keep the robot away from the wet areas. Contact with water will damage the robot.
- To avoid risks of fall, keep your robot in a stable position.
- Do not attach any connectors while robot is powered ON.
- Never leave the robot powered ON when it is not in use.
- Disconnect the battery charger after charging the robot.

Inappropriate Operation:

Inappropriate operation can damage your robot. Inappropriate operation includes, but is not limited to:

- Dropping the robot, running it off an edge, or otherwise operating it in an irresponsible manner.
- Interfacing new hardware without considering compatibility
- Overloading the robot above its payload capacity.
- Exposing the robot to wet environments.
- Continuing to run the robot after hair, yarn, string, or any other item has become entangled in the robot's axles or wheels.
- All other forms of inappropriate operation.
- Using robot in areas prone to static electricity.
- Read carefully paragraphs marked with ⚠ caution symbol.

2. Fire Bird V ATMEGA2560 Omnidirectional Robot

Important: Use Fire Bird V ATMEGA2560 Hardware and Software manual along with this application note.

The Fire Bird V robot is the 5th in the Fire Bird series of robots. First two versions of the robots were designed for the Embedded Real-Time Systems Lab, Department of Computer Science and Engineering, IIT Bombay. Theses platforms were made commercially available form the version 3 onwards. All the Fire Bird V series robots share the same main board and other accessories. Different family of microcontrollers can be added by simply changing top microcontroller adaptor board. Fire Bird V supports ATMEGA2560 (AVR), P89V51RD2 (8051) and LPC2148 (ARM7) microcontroller adaptor boards. This modularity in changing the microcontroller adaptor boards makes Fire Bird V robots very versatile. User can also add his own custom designed microcontroller adaptor board.

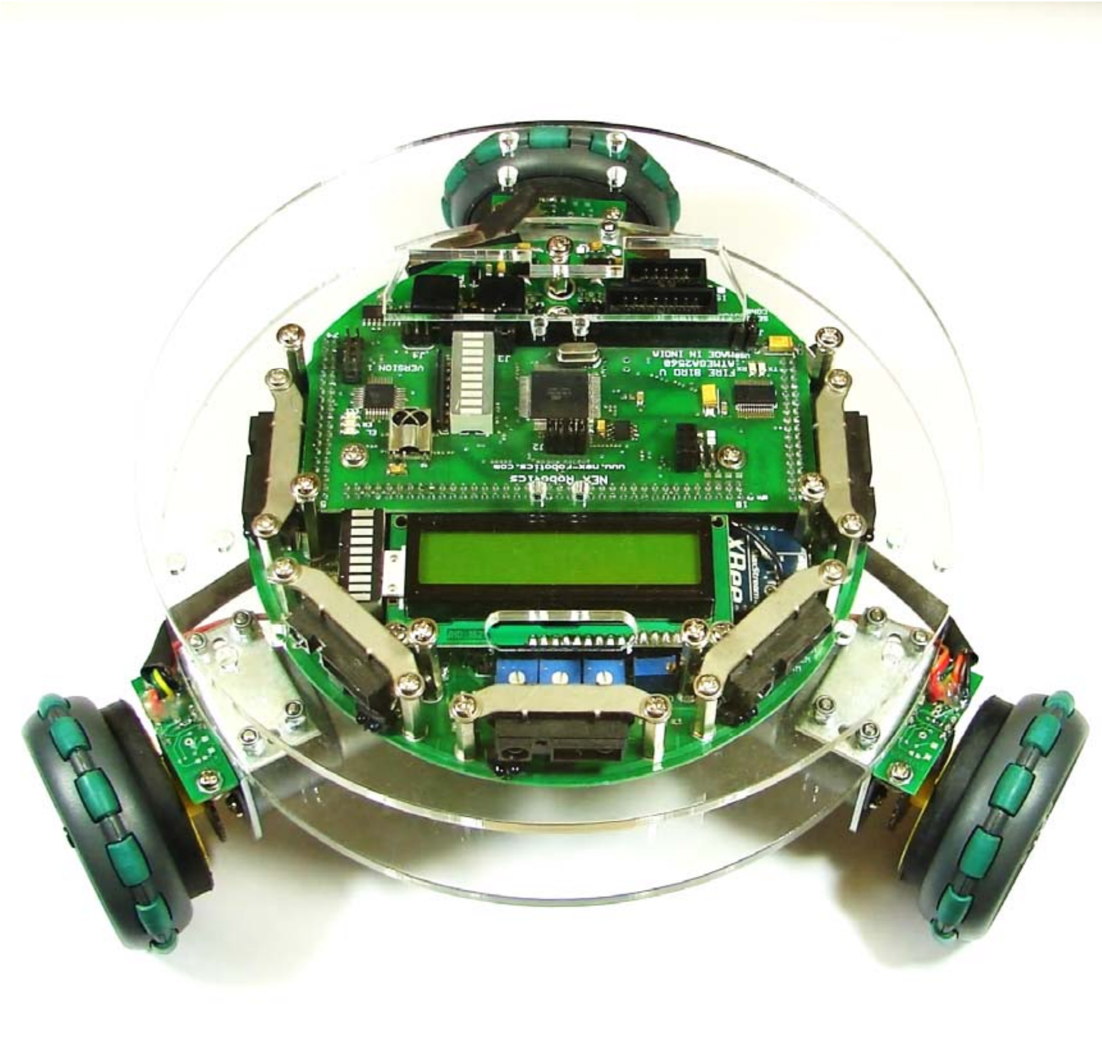


Figure 2.1: Fire Bird V Omnidirectional Robot

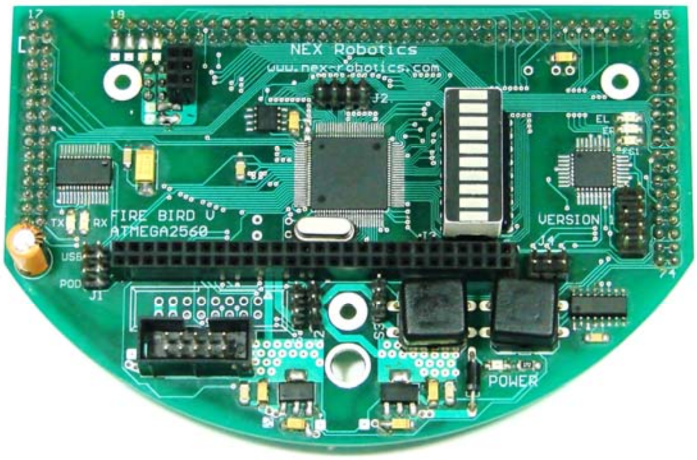


Figure 2.2: ATMEGA2560 (AVR) microcontroller adaptor board on Fire Bird V Omnidirectional Robot

2.1 Fire Bird V Block Diagram:

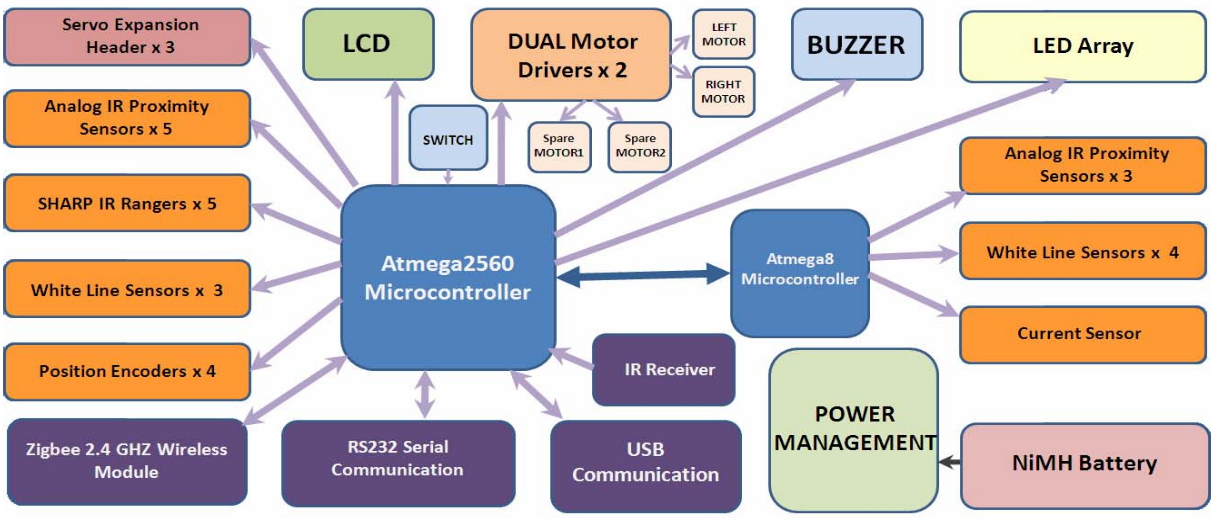


Figure 2.3: Fire Bird V ATMEGA2560 robot block diagram

2.2 Motion control

Fire Bird V ATMEGA2560 Omnidirectional robot uses three DC geared motors with the omnidirectional wheels for the locomotion. It can move in any direction, at any angle with out rotating itself.

Figure 2.4: Omnidirectional Wheel used in the Fire Bird V Omnidirectional Robot

Omnidirectional wheels are wheels with small discs around the circumference which are perpendicular to the rolling direction. The effect is that the wheel will roll with full force, but will also slide laterally with great ease. These wheels are often employed in holonomic drive systems.

They are often used in small robots. In leagues such as Robocon, Robocup etc. Many robots use these wheels to have the ability to move in all directions. Omni wheels are also sometimes employed as powered casters for differential drive robots to make turning faster. However, this design is not commonly used as it leads to Fishtailing.

Robot’s motors are controlled by L293D motor driver IC from ST Microelectronics. To change the direction of the motor, appropriate logic levels (High/Low) are applied to L293D’s direction control pins. Velocity control is done using pulse width modulation (PWM). Single L293D can drive 2 DC motors. Fire Bird V has onboard two L293D ICs to drive four DC motors. Each channel of the motor driver IC can provide current up to 600mA to drive motor.

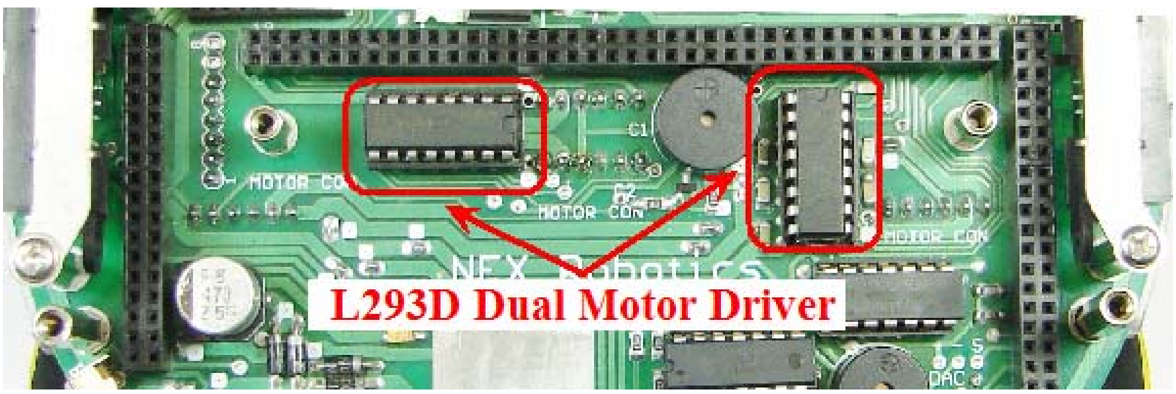


Figure 2.5: Motor Drivers

Pulse Width Modulation for velocity control:
Pulse width modulation is a process in which duty cycle of constant frequency square wave is modulated to control power delivered to the load i.e. motor.

Duty cycle is the ratio of **'T-ON/ T'**. Where **'T-ON'** is ON time and **'T'** is the time period of the wave. Power delivered to the motor is proportional to the **'T-ON'** time of the signal. In case of PWM the motor reacts to the time average of the signal.

PWM is used to control total amount of power delivered to the load without power losses which generally occur in resistive methods of power control.

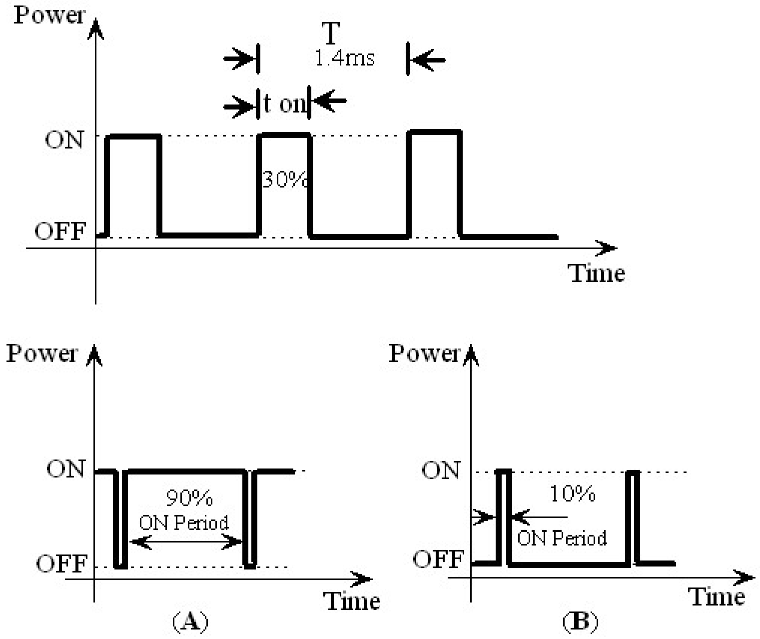


Figure 2.6: Pulse Width Modulation (PWM)

Above figure shows the PWM waveforms for motor velocity control. In case (A), ON time is 90% of time period. This wave has more average value. Hence more power is delivered to the motor. In case (B), the motor will run slower as the ON time is just 10% of time period.

For the Fire Bird V ATMEGA2560 Omnidirectional robot, logic level for the motor direction control is given in the table 3.3.

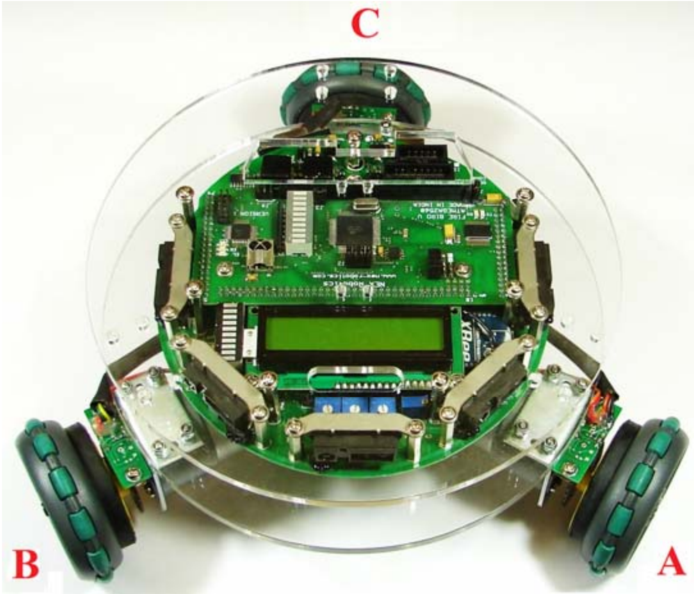


Figure 2.7 Three wheels of the Omnidirectional robot

Microcontroller Pin	Function
PL3 (OC5A)	Pulse width modulation for motor A (velocity control)
PL4 (OC5B)	Pulse width modulation for motor B (velocity control)
PE3 (OC3A)	Pulse width modulation for motor C (velocity control)
PA0	Motor A direction control
PA1	Motor A direction control
PA2	Motor B direction control
PA3	Motor B direction control
PA6	Motor C direction control
PA7	Motor C direction control

Table 2.1: Pin functions for the motion control

We can see all the commands given on the Bargraph LED display which is located at the top right side on the robot. Figure 3.11 shows the location and function of indicator LEDs related to motion control.

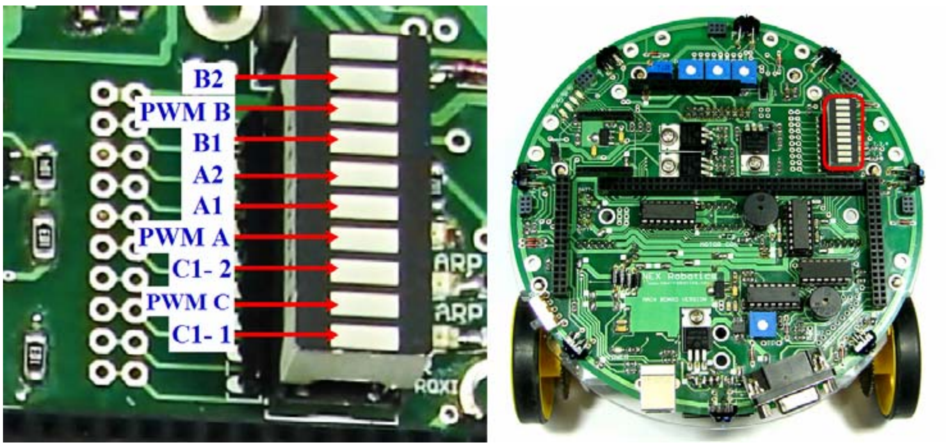


Figure 2.8: Motion status LED indication for ATMEGA2560 microcontroller adaptor board for Fire Bird V

Important:
Auxiliary power can supply current up to 1 Ampere while Battery can supply current up to 2 Ampere. When both motors of the robot changes direction suddenly without stopping, it produces large current surge. When robot is powered by Auxiliary power which can supply only 1 Ampere of current, sudden direction change in both the motors will cause current surge which can reset the microcontroller because of sudden fall in voltage. It is a good practice to stop the motors for at least 0.5seconds before changing the direction. This will also increase the useable time of the fully charged battery.

2.3 Control of an Omnidirectional Robot

Credites: http://www.societyofrobots.com/robot_omni_wheel.shtml

An omni-wheel robot to translate at a particular angle, each wheel must rotate at a particular rotational velocity and direction. Since the robot is moving at angles, it will need to do trigonometric calculations to determine these wheel speeds. However trigonometric calculations are not always possible, since control of an omni-wheel robot very much depends on your robots computational processing ability. If you are using a PC or palmpilot as the robot controller, your robot could do thousands of trigonometric calculations per second. However most microcontrollers cannot. A microcontroller (such as a PIC or AVR), not doing anything else, can do about three trigonometric calculations per second. But unfortunately your robot also needs to read from sensors, drive motors, interpret data, and many other things. Instead, you will need to use what is called a trigonometric lookup table. Basically you do the trig calculations before hand and save them in memory, and the program just references a list. It wont be 100% accurate, or as elegant in code, but it will be much faster - its a balance of speed and accuracy.

Position control is difficult with an omni-wheel robot. The omni-wheel works on the basis of wheel slippage, so using things like encoders for position will not work accurately. You must have the robot detect it's surroundings to track its motion accurately - IMU's, gyros, visual tracking, etc. will work.

Optimization of actuator control is a little complicated. There are several things that need to be considered:

Angle control:
For an omni-wheel robot to translate at a certain angle, each motor needs to go at a certain speed with relation to the others. The speed doesn,t matter, just the ratios.

Motor speed:
The faster the motors go, the faster the robot goes.

Rotational control:
For an omni-wheel robot to rotate at some particular speed, it must add or subtract equally from the motor speed of each motor.

Maximum motor speed:
Motors can only go so fast, so when you do ratios, and rotational control additions, you must make sure that you are not commanding a motor to go faster than it can.

Global angle:
You may or may not want to control for the angle of the robot to it's surroundings.

Steps in the calculation of Motor speeds:

1. Calculate motor ratios to determine translation angle. Do this using the trigonometric lookup tables.

2. Calculate rotational velocity constant. Just add or subtract some constant to the translation motor speeds to rotate clockwise or counterclockwise.

3. Check and correct to make sure maximum motor speed isn’t exceeded. Suppose maximum speed is 100, but your calculations call for motor speeds at 50, -50, and -110. Well the ratio is 50/-50/-110. It must be X/-X/-100, with an unknown X value, based on a maximum speed of 100. Using algebra, your equation then becomes:

   50/110 = X/100
   X = 45.45 =~46

   The motor speeds become 46/-46/-100/

2.4 Functions for PWM output pin configuration and robot’s velocity control

2.4.1 Functions for PWM output pin configuration (called inside the “port_init()” function)

```
//Function to configure ports to enable robot's motion
void motion_pin_config (void)
{
 DDRA = DDRA | 0xCF;   //Motion control pins set as output
 PORTA = PORTA & 0x30; //Inital value of the motion control pins set to 0
 DDRL = DDRL | 0x18;   //Setting PL3 and PL4 pins as output for PWM generation
 PORTL = PORTL | 0x18; //Setting PL3 and PL4 pins as logic 1
 DDRE = DDRE | 0x08;   //Setting PE3 pin as output for PWM generation
 PORTE = PORTE | 0x08; //Setting PE3 pin as logic 1
}
```

2.4.2 Function for robot’s velocity control

```
//Function for robot velocity control
void velocity (unsigned char A, unsigned char B, unsigned char C)
{
	OCR5AL = (unsigned char)A;
	OCR5BL = (unsigned char)B;
	OCR3AL = (unsigned char)C;
}
```

This function takes velocity for left motor and right motor as input parameter and assigns them to output compare register OCR5A and OCR5B. Channel A is used for left motor and channel B is used for right motor. Since we are using PWM in 8 bit resolution its ok to only loading lower byte of the OCR5A and OCR5B registers.

2.5 Application example for robot motion control

Located in the folder “Experiments \ Fire Bire V Omnidirectional Robot” folder in the documentation CD.

This experiment demonstrates robot velocity control using PWM.

Concepts covered: Use of timer to generate PWM for velocity control

There are two components to the motion control:
1. Direction control using pins PORTA0 to PORTA3 and PORTA6, PORTA7
2. Velocity control by PWM on pins PL3, PL4 and PL3 using OC5A, OC5B and OC3A of timer 5 and 3.

Connections: Refer to table 2.1 for connection details.

Note:
1. Make sure that in the configuration options following settings are done for proper operation of the code

   Microcontroller: atmega2560
   Frequency: 14745600
   Optimization: -Os
   (For more information read section: Selecting proper optimization options below figure 4.22 in the hardware manual)

2. Auxiliary power can supply current up to 1 Ampere while Battery can supply current up to 2 Ampere. When both motors of the robot changes direction suddenly without stopping, it produces large current surge. When robot is powered by Auxiliary power which can supply only 1 Ampere of current, sudden direction change in both the motors will cause current surge which can reset the microcontroller because of sudden fall in voltage. It is a good practice to stop the motors for at least 0.5seconds before changing the direction. This will also increase the useable time of the fully charged battery.

3. Robot Control using 'GUI' for Fire Bird V ATMEGA2560 Omnidirectional Robot

Fire Bird V ATMEGA2560 Omnidirectional robot can be controlled by GUI via serial / USB cable or using XBee wireless module. To control the robot using GUI, via RS232 serial communication / USB communication or wireless communication load appropriate hex file on the robot. GUI works on at the 115200 baud rate.

3.1 Loading firmware on the robot

Step 1:
Following firmware (.hex file) needs to be loaded on the robot depending on the mode of communication used.

RS232 serial communication: "Omni_Serial_GUI.hex"
USB communication: "Omni_USB_GUI.hex"
ZigBee wireless module based communication: "Omni_XBee_GUI.hex"

All these hex files are located in the "GUI and Related Firmware \ FB5 Omni Robot" folder.

For information on how to load hex file, refer to chapter 4 of the hardware manual.

Step 2: Connect serial / USB to serial converter cable between robot and PC or install ZigBee wireless module on the robot and connect wireless ZigBee USB module to the PC.

Step 3: Install GUI software

3.2 Installing GUI

Copy "FIRE BIRD V Atmega2560 Omni Robot setup" folder which is located inside the folder "GUI and Related Firmware \ FB5 Omni Robot"

Click on setup.exe which is located in the "FIRE BIRD V Atmega2560 Omni Robot setup" folder.

Follow rest of the steps as mentioned in the Chapter 7 of the Hardware Manual.

3.3 Using GUI

After successful installation go to Start -> All Programs -> FIRE BIRD V Amega2560 -> FIRE BIRD V Atmega2560 Omni Robot or click on "FIRE BIRD V Atmega2560 Omni Robot" thumbnail on your desktop location, GUI will open.

Follow same steps as mentioned in the chapter 7 from the Hardware Manual to establish the connection.

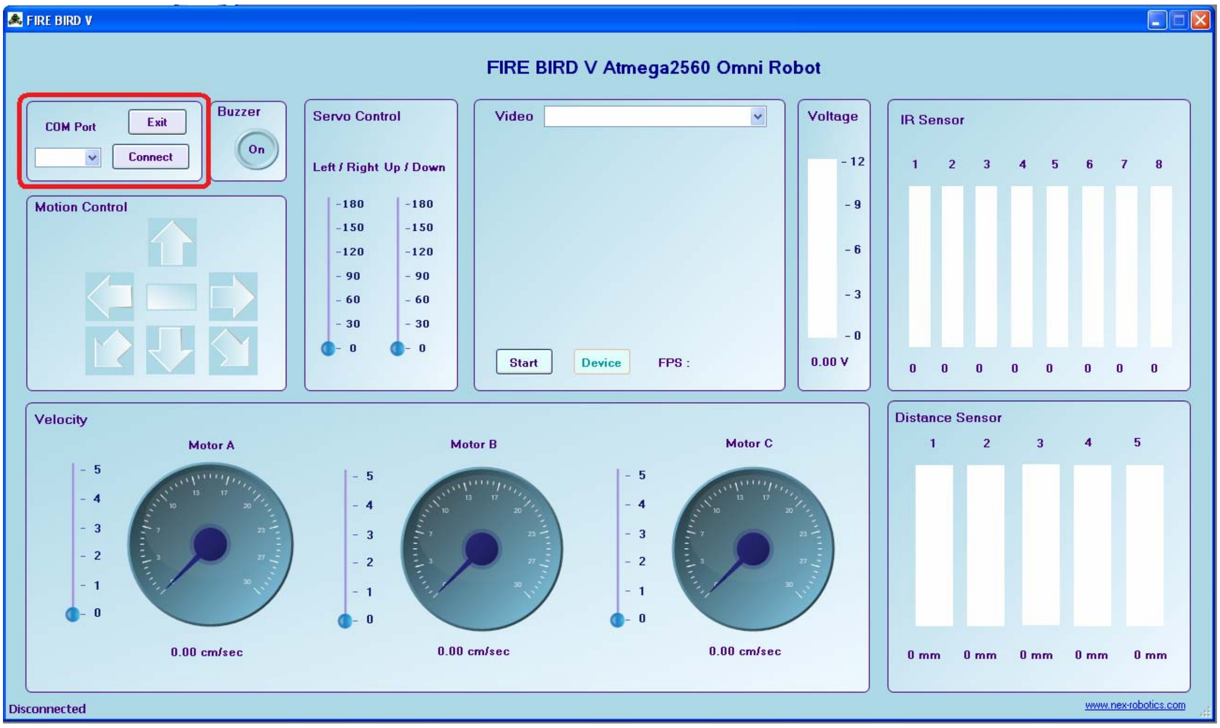


Figure 3.1: Selecting correct com port

⚠**Warning:**
While using USB communication, ensure that the appropriate jumpers are in place. For more details refer chapter 6 of the Hardware Manual.

Note:
For robot connection between robot and PC using USB / serial link or via wireless link, refer to chapter 6 and 7 of the hardware manual.

If you have Wireless camera pod from NEX Robotics and USB TV Tuner card then you can also see the video on the GUI.
For more information on the installation and usage process, refer to documentation of the wireless camera pod.

Follow these steps for video acquisition:

1. Connect USB TV Tuner card with PC and wait for 5 seconds.
2. Start the Fire Bird V robot's GUI
3. In the video window, select devices as USB TV Device. This option will be visible only if USB TV Tuner card is installed and connected.
4. Press start button to acquire the video.

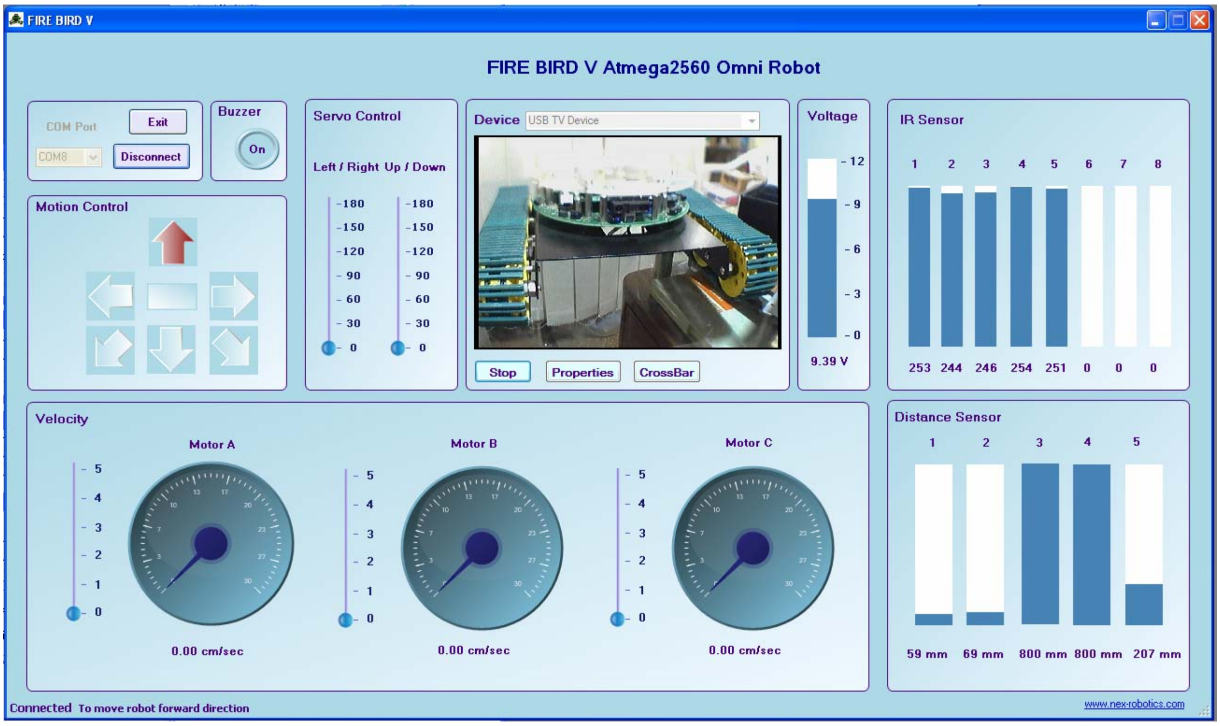


Figure 3.2: Video display on the Fire Bird V robot's GUI