FIRE BIRD V

LPC2148 ARM7
ROBOTIC RESEARCH PLATFORM
Hardware Manual

© IIT Bombay & NEX Robotics Pvt. Ltd.

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

HARDWARE MANUAL

Version 1.0
November 22, 2010

Documentation author
Rohit Chauhan, Nex Robotics Pvt. Ltd.
Sachitanand Malewar, NEX Robotics Pvt. Ltd.
Anant Malewar, NEX Robotics Pvt. Ltd.

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
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
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
Rajanikant Sawant, NEX Robotics
Saurabh Bengali, RA, CSE, IIT Bombay
Vaibhav Daghe, RA, CSE, IIT Bombay
Vibhooti Verma, CSE, M.Tech, IIT Bombay

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

Content of this manual is released under the Creative Commence cc by-nc-sa license. For legal information refer to: http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode

⚠ **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.

Index

1.	**Introduction**	**6**
2.	**Fire Bird V ARM7 LPC2148**	**7**
3.	**Using Fire Bird V Robot**	**11**
4.	**Programming the Fire Bird V ARM7 LPC2148 Robot**	**31**
5.	**Pin Functionality**	**68**
6.	**PC Based Control Using Serial Communication**	**71**
7.	**Robot Control using 'GUI' for Fire Bird V ARM7 LPC2148**	**86**
8.	**Hardware Description**	**93**
9.	**Assembling the Robot and its Accessories**	**126**

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 ARM7 LPC2148

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 of Department of Computer Science and Engineering, IIT Bombay. Theses platforms were made commercially available from 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. At present Fire Bird V supports ATMEGA2560 (AVR), LPC2148 (ARM7) and P89V51RD2 (8051) microcontroller adaptor boards. This feature makes Fire Bird V robots very versatile. User can also add his own custom designed microcontroller adaptor board.

Figure 2.1: Fire Bird V Robots

Figure 2.2: ATMEGA2560 (AVR), LPC2148 (ARM7) and P89V51RD2 (8051) microcontroller adaptor board

Figure 2.3 Fire Bird V ARM7 LPC2148

2.1 Fire Bird V Block Diagram:

Figure 2.4: Fire Bird V ARM7 LPC2148 robot block diagram

2.2 Fire Bird V ARM7 LPC2148 technical specification

Microcontroller:
NXP's ARM7 LPC2148 as Master microcontroller (ARM architecture based Microcontroller)

Sensors:
Three white line sensors
Three Sharp GP2D12 IR range sensor (One in default configuration)
Two analog IR proximity sensors
Two position encoders
Battery voltage sensing
Current Sensing (Optional)

Indicators:
2 x 16 Characters LCD
Indicator LEDs
Buzzer

Control:
Autonomous Control
PC as Master and Robot as Slave in wired or wireless mode

Communication:
Wireless ZigBee Communication (2.4GHZ) (if ZigBee wireless module is installed)
Wired RS232 (serial) communication
Simplex infrared communication (From infrared remote to robot)

Dimensions:
Diameter: 16cm
Height: 10cm
Weight: 1300gms

Power:
9.6V, 2100mAh Nickel Metal Hydride (NiMh) battery pack and external Auxiliary power using battery charger.

Battery Life:
2 Hours while motors are operational at 75% of time

Locomotion:
Two DC geared motors in differential drive configuration and caster wheel at front as support

- Top Speed: 24 cm / second
- Wheel Diameter: 51mm
- Position encoder: 30 pulses per revolution
- Position encoder resolution: 5.44 mm

3. Using Fire Bird V Robot

In this chapter various components of the robot and there principal of operations are explained in brief. For more details user can refer to Chapter 8, which covers robot's hardware in detail. It is very important that user goes through chapter 3 and 8 before starting to use robot.

3.1 Powering up Fire Bird V

Fire Bird V has onboard rechargeable 9.6V, 2.1Ah Nickel Metal Hydride battery which can power the robot for approximately 2 hours. In case experiments are to be performed for an extended period robot can also be powered by external, auxiliary power supply.

Figure 3.1: Connecting the battery on the Fire Bird V main board

For the safety during transportation, robot's battery is disconnected. Before connecting battery to the robot, make sure that power switch is moved towards back direction of the robot (AP). You need to charge the battery before first use. Refer to section 3.3 for battery charging. For running the robot on battery power or auxiliary power, refer to the section 3.4 and 3.5.

Figure 3.2: Battery Connector

Figure 3.3: Power Switch

3.2 Power management system on the Fire Bird V

Fire Bird V is powered by 9.6V, 2.1Ah rechargeable Nickel Metal Hydride battery pack. When fully charged, battery pack gives 11.5V. When it is fully discharged, voltage drops to about 8.5V. Battery pack should not be discharged below 8V (1V per cell) to extend the battery life. Nickel Metal Hydride batteries must be recharged using smart charging circuit which follows the appropriate charging profile for the batteries. To avoid any damage to the batteries, only use charger provided with the robot.

Power management block on the Fire Bird V performs following functions.

1. Battery voltage monitoring
2. Battery current sensing*
3. Battery low warning in case battery voltage is below critical level
4. Regulated supply for onboard payload.

*Current sensing is an optional accessory.

Battery low threshold value is factory set at 8V. Battery low threshold value can be adjusted by turning potentiometer which is shown in Figure 3.4. Robot will start giving beeping sound when battery is critically low and needs recharge.

Figure 3.4: Power Management section on the main board

3.3 Battery Charging

Fire Bird V is powered by 9.6V, 2.1Ah NiMH rechargeable battery. NiMH charger provided with the robot will charge the battery in 4.5 hours. Battery has built-in thermistor for monitoring battery temperature during battery charging.

Figure 3.5: Smart NiMH battery charger for Fire Bird V

Figure 3.5 shows the smart battery charger for Fire Bird V robot. Battery charger checks state of battery before initiating the charging process. If battery is deeply discharged it preconditions the battery before starting full charging cycle. While charging the battery, by looking at battery voltage, current and temperature it selects optimal charging algorithm.

Figure 3.6: Connection for battery charging

Battery charging procedure:

1. Turn off the robot.
2. Connect AC adaptor in the mains. Connect AC adaptor's output to the battery charger's "AC Adaptor jack" connector as shown in figure 3.5 and turn on the adaptor. Battery charger's power LED will turn on and red and green LEDs will blink twice.
3. Insert one end of the connector which is marked with the label "Battery Charging" in to the socket which is located at the back side of the robot as shown in the figure 3.6. Insert the other end of the connector in to "Battery Connector 1" of the battery charger as shown in the figure 3.5. To start the battery charging, press button marked with "Start Push Button". After a small delay green LED will turn on which indicates battery is being charged.
4. When battery is fully charged red and green status LEDs will blink alternately.
5. If there is any fault then charger will give different types of LED indication to indicate nature of fault. Refer to table 3.1 for interpretation of the LED status.

Indicator LEDs		**Battery Status**	**Charger Status**
Green	Red		
On	Off	Voltage < 11.4 V	Battery is getting charged
Off	Off	Charger idle	Charger is in idle state. Press "Start push button" to start charging
Off	On	Fault	Charge termination due to over current
On	On	Fault	Charge termination due to timeout
Alternate blinking		Fully charged	Battery is fully charged

Table 3.1 Battery charging indicator LED status

Note:

- If fault condition occurs, then wait for some time till battery is cooled down and start recharging again after some time. If problem persists then replace the battery.
- While battery is being charged robot cannot be turned ON irrespective of the position of the power switch.

3.4 Powering the robot on battery power

To turn ON the robot on the battery power, make sure that battery is connected to the robot and move the power switch towards forward direction (BP) as shown in the figure 3.3

3.5A Powering the robot on auxiliary power

Usually robot is powered by onboard battery. In order to do experiments for longer duration without worrying about the battery getting low, robot can be powered by external power source which is also called as auxiliary power source. Auxiliary power source provides regulated 12V, 1Amp. supply to the robot.

Figure 3.7: Robot Powering via Battery and Auxiliary supply

To turn ON the robot on the Auxiliary power use following steps:

1. Disconnect the battery by removing the battery connector on the main board which is located at the bottom side of the robot.
2. Move the power switch to the BP (refer to figure 3.3).
3. Insert the connector marked with the “Auxiliary Power” in the connector which is located at the back side of the robot as shown in the figure 3.7. Connect the other end of the connector to the “Battery Connector 1” on the battery charger (refer to figure 3.5) Connect the AC adaptor to the battery charger unit and turn on the power.
4. Now to turn ON the robot move power switch towards “AP” (backwards) (figure 3.3).

⚠**Warning:**

Do not connect auxiliary power while battery is connected to the robot. In such case robot will either run on the battery if switch is in Battery Power mode (“BP” or forward direction) or in auxiliary power mode (“AP” or backward direction). Robot will not turn off.

3.5B Battery Maintenance

Fully charged NiMh battery will get completely discharged within a week. It is always recommended to charge batteries before using the robot. If battery is charged after for about a month in the first charge it can deliver only 1/3rd power of its rating. To restore battery to its full potential again, perform at least 2-3 charge discharge cycles.

To ensure long life, charge battery at least once a week and discharge it till robot starts giving battery low warning. Repeat this cycle for at least 2-3 times. For faster battery discharge load "Motion_Control_Simple" program which is located in the "Experiments" folder in the documentation CD. Loading the program on the robot is covered in the chapter 4 in the hardware manual or chapter 2 in the software manual.

3.6 Motion control

FIRE BIRD V robot has two DC geared motors for the locomotion. The robot has a top speed of 24cm/second. These motors are arranged in differential drive configuration, i.e. motors can move independently of each other. Front castor wheel provides support at the front side of the robot. Using this configuration, the robot can turn with zero turning radius by rotating one wheel in clockwise direction and other in counterclockwise direction. Position encoder discs are mounted on both the motor's axle to give a position feedback to the microcontroller.

Figure 3.8: Motion Control

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.

*Refer Comparison chart for LPC2148 compatibility.

Figure 3.9: 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 3.10: 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 ARM7 LPC2148 version, logic level for the motor direction control is given in the table 3.3.

Microcontroller Pin	Function
P0.7 (PWM2)	Pulse width modulation for the left motor (velocity control)
P0.21 (PWM5)	Pulse width modulation for the right motor (velocity control)
P0.22	Left motor direction control
P1.21	Left motor direction control
P0.10	Right motor direction control
P0.11	Right motor direction control

Table 3.2: Pin functions for the motion control

ROBOT DIRECTION	LEFT BWD (LB) P0.22	LEFT FWD(LF) P1.21	RIGHT FWD(RF) P0.10	RIGHT BWD(RB) P0.11	PWM P0.7 for left motor P0.21 for right motor
FORWARD	0	1	1	0	As per velocity requirement
REVERSE	1	0	0	1	As per velocity requirement
RIGHT *(Left wheel forward, Right wheel backward)*	0	1	0	1	As per velocity requirement
LEFT*(Left wheel backward, Right wheel forward,)*	1	0	1	0	As per velocity requirement
SOFT RIGHT*(Left wheel forward,, Right wheel stop)*	0	1	0	0	As per velocity requirement
SOFT LEFT*(Left wheel stop, Right wheel forward,)*	0	0	1	0	As per velocity requirement
SOFT RIGHT 2 *(Left wheel stop, Right wheel backward)*	0	0	0	1	As per velocity requirement
SOFT LEFT 2 *(Left wheel backward, Right wheel stop)*	1	0	0	0	As per velocity requirement
HARD STOP	0	0	0	0	As per velocity requirement
SOFT STOP (Free running stop)	X	X	X	X	0

Table 3.3: Logic table for motor direction 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 3.11: Motion status LED indication for ARM7 LPC2148 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 robot is on auxiliary power and robot changes is direction suddenly from forward to backward or vice versa, it will cause large current surge and robot's microcontroller might get reset. This will happen repeatedly and you might think that problem is in the software. To avoid this problem use stop command for at least 0.5 second before changing direction. This problem will not occur when robot is on battery power and batteries have sufficient charge. But still it is a good practice to stop robot for 0.5 seconds before changing the direction to increase the life of the motor.

3.7 Position Encoders

Position encoders give position / velocity feedback to the robot. It is used in closed loop to control robot's position and velocity. Position encoder consists of optical encoder and slotted disc assembly. When this slotted disc moves in between the optical encoder we get square wave signal whose pulse count indicates position and time period indicates velocity.

Optical encoder MOC7811 is used for position encoder on the robot. It consists of IR LED and the photo transistor mounted in front of each other separated by a slot in the casing. When IR light falls on the photo transistor it gets in to saturation and gives logic 0 as the output. In absence of the IR light it gives logic 1 as output. A slotted encoder disc is mounted on the wheel is placed in between the slot. When encoder disc rotates it cuts IR illumination alternately because of which photo transistor gives square pulse train as output. Output from the position encoder is cleaned using Schmitt trigger based inverter (not gate) IC CD40106.

Figure 3.12: DC geared motors and position encoders

Figure 3.13: Position encoder assembly

Calculation of position encoder resolution:

Case 1: Robot is moving forward or backward (encoder resolution is in mm)

Wheel diameter: 5.1cm
Wheel circumference: 5.1cm * 3.14 = 16.014cm = 160.14mm
Number slots on the encoder disc: 30
Position encoder resolution: 163.2 mm / 30 = 5.44mm / pulse.

Case 2: Robot is turning with one wheel rotating clockwise while other wheel is rotating anti clockwise. Center of rotation is in the center of line passing through wheel axel and both wheels are rotating in opposite direction (encoder resolution is in degrees)

Distance between Wheels = 15cm

Radius of Circle formed in 360^0 rotation of Robot = Distance between Wheels / 2

= 7.5 cm

Distance Covered by Robot in 360^0 Rotation = Circumference of Circle traced
= 2 x 7.5 x 3.14
= 47.1 cm or 471mm

Number of wheel rotations of in 360^0 rotation of robot
= Circumference of Traced Circle / Circumference of Wheel
= 471 / 160.14
= 2.941

Total pulses in 360^0 Rotation of Robot
= Number of slots on the encoder disc / Number of wheel rotations of in 360^0 rotation of robot
= 30 x 2.941
= 88.23 (approximately 88)

Position Encoder Resolution in Degrees = 360 / 88
= 4.090 degrees per count

Case 3: Robot is turning with one wheel stationary while other wheel is rotating clockwise or anti clockwise. Center of rotation is center of the stationary wheel (encoder resolution is in degrees)

In this case only one wheel is rotating and other wheel is stationary so robot will complete its 360^0 rotation with stationary wheel as its center.
Radius of Circle formed in 360^0 rotation of Robot = Distance between Wheels
= 15 cm

Distance Covered by Robot in 360^0 Rotation = Circumference of Circle traced
= 2 x 15 x 3.14
= 94.20 cm or 942 mm

Number of wheel rotations of in 360^0 rotation of robot
= Circumference of Traced Circle / Circumference of Wheel
= 942 / 160.14
= 5.882

Total pulses in 360^0 Rotation of Robot
= Number of slots on the encoder disc / Number of wheel rotations of in 360^0 rotation of robot
= 30 x 5.882
= 176.46 (approximately 176)

Position Encoder Resolution in Degrees = 360 /176
= 2.045 degrees per count

Figure 3.13: Position encoder pulse LEDs on ARM7 LPC2148 microcontroller adaptor board

Pulses coming from the position encoders are indicated by position encoder LEDs on the ARM7 LPC2148 board as shown in figure 3.13

3.8 Sharp IR range sensors

For accurate distance measurement, robot uses Sharp IR range sensors. The robot can be fitted with five IR range sensors as shown in figure 3.14.However LPC2148 socket supports only three such sensors mounted at positions 2,3 and 4 on the main board as shown in Figure 3.14. For more details refer comparison. Sharp IR range sensors consists of narrow IR beam for illumination and CCD array, which uses triangulation to measure the distance from any obstacle. A small linear CCD array is used for angle measurement. When light hits the obstacle and reflects back to the linear CCD array. Depending on the distance from the obstacle, angle of the reflected light varies. This angle is measured using the CCD array to estimate distance from the obstacle. It gives same response to different colored objects as measured distance is function of the angle of reflection and not on the reflected light intensity. Figure 3.15 shows the internals of the sensor. Figure 3.16 explains how change in the distance from the obstacle can be measured by measuring angle of reflection of light beam from the obstacle. Since sensor measurement is based on triangulation and not on intensity of the reflected light it is immune to disturbance caused by ambient light.

Sensor gives out analog voltage corresponding to angle of reflection. Relationship between the angle of reflection and output voltage is not linear because of trigonometry involved. Also these sensors cannot cover initial range of 0 to some distance depending on the type of sensor. It is known as blind spot distance. In this range sensor gives incorrect readings. Table 3.4 gives information about sensing range and the blind spot distance for each sensor.

Figure 3.14: Sharp Sensors mounted on Fire Bird V

Figure 3.15: Infrared Range finder sensor and its inside view

Figure 3.16: Distance measurement based on angel of reflection

Fire Bird V supports three types of IR range sensors from Sharp Microelectronics.

GP2D120 GP2D12 GP2Y0A02YK

Figure 3.17: Sharp IR Range sensors for Fire Bird V

Sensor	Range	Blind Spot
GP2D120X	30cm to 20cm	4cm to 0cm
GP2D12	80cm to 10cm	10cm to 0cm
GP2Y0A02	150cm to 20cm	20cm to 0cm

Table 3.4: Sharp IR Range sensors coverage

Figure 3.18: Distance Vs. Output voltage of GP2D120, GP2D12 and GP2Y0A02YK

Figure 3.18 shows the typical output character of the GP2D120, GP2D12 and GP2Y0A02YK sensors. In these graphs X axis represents distance from the obstacle and Y axis represents the output voltage. The sensor's output characteristic is slightly logarithmic in nature hence to get the distance in millimeters we have to use following formulas.

Distance in mm for GP2D120 = 10.00*((1.00/ ((0.001240875*(float) ADC value) + 0.005)) – 0.42)

Distance in mm for GP2D12 =

```
(int)(10.00 * (2799.6 * (1.00 /  (float)((double)(ADC_Value)^(double)(1.1546)))));
```

Figure 3.14 shows locations of the Sharp IR range sensors on the robot. Each sensor consumes about 30mA current. If a sensor emits some signal and senses any object based on the reflected signal then it is known as active sensor. If many robots in the same field are using active sensors such as Sharp IR range sensors, IR proximity sensors or ultrasound sensors then they will interfere with the sensors of the other robots.

To prevent this interference from occurring Fire Bird V can switch all of its active sensors on and off. This means that many Fire Bird series robots can work in same field without causing

interference with other Fire Bird robot's sensor. Many robots can synchronize when particular sensor should be turned on or off over wireless communication using ZigBee wireless modules. Another advantage of this feature is that robot can turn off these sensors when not in use to conserve power.

Note: For more details about the sensor interfacing, turning on /off the sensors, refer to section 8.3 in the chapter 8. It also shows the jumpers for the bypassing the MOSFET switches. You can also check whether particular sensor is on or off by looking at the LEDs which are located just right side of the bar graph LED display.

Figure 3.19: Sensor status indicator LEDs

3.9 Infrared proximity and directional light intensity sensors

Infrared proximity sensors are used to detect proximity of any obstacles near the robot. Directional light intensity sensor gives intensity of the light in particular direction. IR proximity sensor consists of IR LED and IR photo diode. When obstacle comes within sensing range IR photo diode detects reflected light of the IR LED. To reduce interference with ambient light photo diode is cased in material which allows only IR light to pass. While sensing obstacle in order to eliminate interference due to ambient light, reading should be taken while IR LED is on and off. By comparing these two values effect of ambient light can be nullified. To detect directional light intensity, IR LED is turned off and light intensity at photo diode is measured. Fire Bird V has eight IR proximity sensors. Currently LPC2148 supports only two such sensors mounted at position no. 2 and 4 on the main board as shown in Figure 3.20. For more details refer compatibility chart. IR LEDs of all these sensors can be turned on and off by MOSFET Q3. To turn on IR LEDs permanently J4 needs to be shorted. Refer to table 8.4 for more details.

In the absence of the obstacle there is no reflected light hence no leakage current will flow through the photo diode and output voltage will be around 5V. As obstacle comes nearer, more light gets reflected and falls on the photo diode and leakage current flowing through the photo diode starts to increase and voltage across the diode starts to fall.

When IR proximity sensors used in conjunction with Sharp IR Range sensors, IR proximity sensors covers sensing range in which Sharp IR Range sensors have blind spot. IR proximity sensors working together with Sharp IR range sensors covers range from 0 to maximum range of the Sharp IR sensor without having any blind spot.

Figure 3.20: Eight IR proximity sensors on Fire Bird V

3.10 White Line Sensor:

White line sensors are used for detecting white line on the ground surface. White lines are used to give robot sense of localization. White line sensor consists of a highly directional phototransistor for line sensing and red LED for illumination. Due to the directional nature of the phototransistor it does not get affected with ambient light unless it is very bright.

Figure 3.21: White line sensor

Figure 3.22: White Line sensor

Figure 3.23: Schematic of the white line sensor

When the robot is not on a white line, amount of light reflected is less hence less leakage current flows through the photo transistor. In this case, the line sensor gives an output in the range of 2 volts to 5 volts. When the sensor is on a white line, more light gets reflected resulting in considerable increase in the leakage current and sensor's output voltage falls to 0.15 to 0.8 volts.

Power to the red LEDs is controlled by MOSFET which can be turned off to extend robot's battery life. This feature is also very useful in advanced methods of line following in which reflected light intensity from the surface is measured while red LEDs are on and off to nullify the effect of the ambient light on the sensor reading. To turn on red LEDs permanently put jumper on J4. J4 controls power to the red LED of white line sensors 1, 2 and 3 and Sharp sensors 2, 3 and 4. Refer to the table 8.4 for more information on jumper settings.

White line Sensor	ADC channel number
Left	ADC channel no. AD1.3
Center	ADC channel no. AD0.1
Right	ADC channel no. AD0.2

Table 3.5 White line sensor connections with ADC of ARM7 LPC2148

White Line sensor calibration

By using trimming potentiometers located on the top center of the main board, line sensors can be calibrated for optimal performance. Line sensors are factory calibrated for optimal performance. Using these potentiometers we can adjust the intensity of the red LEDs of the white line sensor. Sensitivity adjustment is needed when color contrast between the white and non-white surface in a white line grid is sub-optimal. In such cases the sensors can be tuned to give maximum difference between white and non white surfaces. You can also turn on and turn off IR LEDs and take sensor readings at the same place and nullify the effect of the ambient light.

Figure 3.24: Potentiometers for white line sensor calibration

3.11 LCD Interfacing

Figure 3.25: LCD socket and other settings

Figure 3.26: LCD socket pin connection

Figure 3.27 LCD interfacing with the microcontroller

Microcontroller	LCD PINS	Description
VCC	VCC	Supply voltage (5V).
GND	GND	Ground
P1.19	RS (Control line)	Register Select
P1.18	R/W (Control line)	READ /WRITE
P1.17	EN (Control Line)	Enable
P1.22 to P1.25	D4 to D7 (Data lines)	Bidirectional Data Bus
	LED+, LED-	Backlight control

Table 3.6 LCD Pin mapping and functions

To interface LCD with the microcontroller requires 3 control signals and 8 data lines. This is known as 8 bit interfacing mode which requires total 11 I/O lines. To save number of I/Os required for LCD interfacing we can use 3 control signals with 4 data lines. This is known as 4 bit interfacing mode and it requires 7 I/O lines. We are using 4 bit interfacing mode to reduce number of I/O lines. In this mode higher nibble and lower nibble of commands/data set needs to be sent separately. Figure 3.27 shows LCD interfacing in 4 bit mode. The three control lines are referred to as EN, RS, and RW.

The EN line is called "Enable" and it is connected to P1.17. This control line is used to tell the LCD that microcontroller has sent data to it or microcontroller is ready to receive data from LCD. This is indicated by a high-to-low transition on this line. To send data to the LCD, program should make sure that this line is low (0) and then set the other two control lines as required and put data on the data bus. When this is done, make EN high (1) and wait for the minimum amount of time as specified by the LCD datasheet, and end by bringing it to low (0) again.

The RS line is the "Register Select" line and it is connected to P1.19. When RS is low (0), the data is treated as a command or special instruction by the LCD (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is treated as text data which should be displayed on the screen.

The RW line is the "Read/Write" control line and it is connected to P1.18. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading from) the LCD.

The data bus is bidirectional, 4 bit wide and is connected to P1.22 to P1.25 of the microcontroller. The MSB bit (DB7) of data bus is also used as a Busy flag. When the Busy flag is 1, the LCD is in internal operation mode, and the next instruction will not be accepted. When RS = 0 and R/W = 1, the Busy flag is output on DB7. The next instruction must be written after ensuring that the busy flag is 0.

Figure 3.28: LCD Timing Diagram.

3.12 Buzzer

Robot has 3 KHz piezo buzzer. It can be used for debugging purpose or as attention seeker for a particular event. The buzzer is connected to P0.25 pin.

Figure 3.29: Buzzer

4. Programming the Fire Bird V ARM7 LPC2148 Robot

The LPC2148 microcontroller is supported by various commercially available IDEs for compiling and debugging of the code. Keil being one of them is the widely used IDE for LPC family of microcontrollers. The µVision4 IDE is Windows-based software development platforms that combines a robust editor, project manager, and make facility. µVision4 integrates all tools including the C compiler, macro assembler, linker/locator, and HEX file generator. The evaluation version of Keil µVision4 IDE is used for demonstrating the sample codes that are included in the CD. The evaluation version has the code size limit of 32KB above which it is restricted to compile the code. The open source community has been doing a lot in the development of open source tools for ARM architecture based microcontrollers. The open source tools are available at zero cost and are being improved with time. Eclipse being one of them and is most commonly used IDE due to its unique features like auto complete, project tree, etc. It requires GCC tool chain for code compilation.

The LPC series of microcontrollers are preloaded with the boot loader firmware which allows self programming of microcontrollers using serial port. Flash magic is a utility which provides an interface for reading, writing and verifying the flash memory of the microcontroller.

4.1 Installing Keil µVision4 IDE

To install **Keil µVision4 IDE**, Go to "Software" folder in the documentation CD and locate "**mdk412.exe"** file. Click on "**mdk412.exe**" to start the installation process. Once the installation process is started Keil µVision welcome screen will appear.

Please read the instructions on the welcome window and click **Next>>** to start the installation.

Figure 4.1

Please read the license agreement carefully. If it is acceptable click the check box and click **Next>>** to continue.

Figure 4.2

Select the destination folder where setup will install files. It is always recommended to select the default location. To create backup of old installation select the backup option and click **Next>>** to continue.

Figure 4.3

In the next window enter your information and click **Next>>** to continue.

Figure 4.4

On clicking next the file copying process will begin. Wait till setup is complete.

Figure 4.5

Click **Finish** to complete installation process.

Figure 4.6

4.2 Installing Flash Magic Tool

To install **Flash Magic**, Go to "Software" folder in the documentation CD and locate "**FlashMagic.exe"** file. Click on "**FlashMagic.exe**" to start the installation process. Once the installation process is started Flash Magic welcome screen will appear.

Figure 4.7

Please read the license agreement carefully. If it is acceptable click the radio button and click **Next>>** to continue.

Figure 4.8

Select the destination folder and click **Next>** to continue.

Figure 4.9

In the next window choose the appropriate folder and click **Next>** to continue.

Figure 4.10

Select the desired option and click **Next>** to continue.

Figure 4.11

Setup is now ready to begin installing flash magic. Click **Install** to continue.

Figure 4.12

Wait till setup installs Flash Magic on your computer.

Figure 4.13

Click Finish to complete the installation process

Figure 4.14

4.3 Overview of Keil µVision4 IDE

To start Keil IDE click Start>Programs>Keil µVision4. The initial screen will appear followed by the main window.

Figure 4.15

The Keil IDE main window in basic configuration is mainly divided into three areas.

Editor - It is the area where .c and .h files of the project are edited.
Project Explorer- It shows the project tree.
Output Window- This window shows the messages related to compiling, project building and debugging.

4.3.1 Setting up Project in Keil uVision

1. To create a new project, Select **Project>New uVision Project** from the main menu.

Figure 4.16

2. Create a new directory and name it as **First_Project**. Click open to enter in to this directory.

Figure 4.17

3. Inside this directory create a new project and name it as **First_Project** and click **Save** to continue.

Figure 4.18

4. In the next window locate **NXP (founded by Philips)** tree and expand it.

Figure 4.19

5. Now select target device as **LPC2148** and click **OK** to continue.

Figure 4.20

6. Click **Yes** to copy **Startup.s** file to project folder. This file configures stack, PLL and maps memory as per the configurations in the wizard. It is discussed in the later sections.

Figure4.21

7. Observe the project explorer area in the main window.

Figure 4.22

8. Now click **Project>Manage>Components, Environments, Books** from the main menu to ensure compiler settings.

Figure 4.23

9. In the **Folders/Extensions** tab ensure the compiler settings are as shown in the Figure below. If you have installed keil software at a different location then change **Tool Base Folder** location. Click **OK** to continue.

Figure 4.24

10. Now click **File>New** to create a new file.

Figure 4.25

11. Save the new file in to the same folder that was created earlier and name it as **main.c** and click **Save** to continue.

Figure 4.26

12. Now add "**main.c**" to the source group by right clicking on the **Source Group 1** from the project explorer and select the highlighted option as shown in the Figure below.

Figure 4.27

13. Select "**main.c**" file to be added and click **ADD** to continue.

Figure 4.28

14. Observe that **"main.c"** file is added to the source group in the project explorer window.

Figure 4.29

15. Right click **Target1** in the project explorer window and select the highlighted option as shown in the Figure below.

Figure 4.30

16. In the appearing window select **Target** tab and set Xtal. frequency as 12MHz.

Figure 4.31

17. In the **Output** tab ensure that **Create HEX File** option is selected.

Figure 4.32

18. In the **Linker** tab ensure that the highlighted option is selected and click **OK** to continue.

Figure 4.33

4.4 Writing your first code in Keil

Now since the project is almost setup we can start writing code in the "**main.c**" file that was created earlier. For demonstration purpose you can copy the following code and paste it in the "**main.c**" file. This code is written to blink bargraph LEDs on the robot in the sequential order starting from top to bottom.

```
#include  <lpc214x.h>


/**********Macros******************/

#define BUZZER_OFF() IO0CLR=(1<<25)          //Macro to turn OFF buzzer
#define BUZZER_ON() IO0SET=(1<<25)           //Macro to turn ON buzzer

/***********************************/


/****Function Prototypes**********/

void Buzzer_Delay(void);
void Init_Buzzer_Pin(void);
void Init_Peripherals(void);
void Init_Ports(void);

/***********************************/


/*************************************************************

      Function          : Buzzer_Delay
      Return type       : None
      Parameters        : None
      Description       : Provides small amount of delay between
                                buzzer toggles.
*************************************************************/
void Buzzer_Delay(void)
{
 unsigned int  i,j;
 for(j=0;j<20;j++)
 {
  for(i=0; i<60000; i++);
 }
}


/*************************************************************

      Function          : Init_Buzzer_Pin
      Return type       : None
      Parameters        : None
      Description       : Initialises Buzzer pin
*************************************************************/
```

```
void Init_Buzzer_Pin(void)
{
 PINSEL1&=0xFFF3FFFF;
 PINSEL1|=0x00000000;          //Set P0.25 as GPIO
 IO0DIR&=0xFDFFFFFF;
 IO0DIR|= (1<<25);             //Set P0.25 as Output
 BUZZER_OFF();                       //Initially turn OFF buzzer
}


void Init_Ports(void)
{
 Init_Buzzer_Pin();
}

void Init_Peripherals(void)
{
 Init_Ports();
}

int main(void)
{
 PINSEL0 = 0x00000000;         // Reset all pins as GPIO
 PINSEL1 = 0x00000000;
 PINSEL2 = 0x00000000;
 Init_Peripherals();         // Init Port pins and Peripherals
 while(1)
 {
  BUZZER_ON();                             //Turn ON buzzer
  Buzzer_Delay();                    //Wait
  BUZZER_OFF();                            //Turn OFF Buzzer
  Buzzer_Delay();                    //Wait
 }
}
```

We are now going to compile this code to generate the hex file which we will load on the Robot's microcontroller. To build the project click on **Rebuild** button on the main toolbar.

Figure 4.34

You can alternatively build project by clicking on **Project>Build Target** from the main menu.

Figure 4.35

You can observe the build process in the output window. If any errors, rectify it by double clicking on it and you will be pointed to the erroneous line.

Figure 4.36

You can also verify that "first_project.hex" file is generated in the folder you have selected.

4.5 Code Walkthrough

In this code, first line represents processor specific header file declaration. The # include directive is used for including header files in the existing code. These file contains the definitions of special function registers related to this device. This file can be found in \ Keil\ARM\INC\Philips subfolder of **Keil** installation directory.

The next two lines,

```
#define BUZZER_OFF() IO0CLR=(1<<25)     //Macro to turn OFF buzzer
#define BUZZER_ON() IO0SET=(1<<25)      //Macro to turn ON buzzer
```

declares macros to turn ON and OFF buzzer. It mainly controls the state of pin (P0.25) that is connected to buzzer.

The function `Buzzer_Delay(x)` generates app 1 mSec delay. The integer value passed to this function determines the delay period.
In all the codes we will configure pins related to any particular module in the `Init_xxxx_Pin()` functions. In this example code we have used the function `Init_Buzzer_Pin`. Buzzer is connected to P0.25 pin of the microcontroller. P0.25 is configured as output with the initial value set to logic 1 to keep buzzer off at the time of port initialization. All the `Init_xxxx_Pin()` functions will be initialized in the `Init_Peripherals()` function in all the codes as a convention.

In the above code buzzer is turned on by calling macro `BUZZER_ON()`
`Buzzer_Delay(1000)` introduces delay of approximately 1 second.
Buzzer is turned off by calling macro `BUZZER_OFF()`

`Buzzer_Delay(1000)` introduces delay of approximately 1 second.

All these statements are written in `while(1)` loop construct to make buzzer on and off periodically.

4.6 Debugging the code in uVision in Simulator mode

To use Debugger in simulator mode first we need to setup the debugger so as to use simulator.
1. Right click **Target 1** and select the highlighted option as shown in the figure below.

Figure 4.37

2. In the debug tab make sure that the highlighted settings are done. Click **OK** to continue.

Figure 4.38

3. Start Debug session by clicking on **Debug>Start/Stop Debug Session** from the main menu. Alternatively you can press Ctrl+F5 to toggle between starting and stopping of debug session.

Figure 4.39

4. Click **OK** to continue.

Figure 4.40

5. Go to **Peripherals>System Control Block>Phase Locked Loop 0** to observe the PLL settings on the microcontroller.

Figure 4.41

6. In the **Crystal Oscillator & Processor Clock** section observe that XTAL is 12MHz and CCLK is 60MHz. This is because the debugger has already executed the start up code when we started the debug session. It also ensures that PLL settings that we had done in the configuration wizard of the startup code are properly working.

Figure 4.42

7. Now click **Peripherals>Pin Connect Block** to include pin connect block,

Figure 4.43

8. Similarly select **Port 1 window.**

Figure 4.44

9. Arrange the windows as shown in figure below.

Figure 4.45

10. Now use **Step** function to step through the code. Alternatively you can press Ctrl+F11. Observe the changes in the Pin connect block and Port1 block as you step through the code.

Figure 4.46

11. When you encounter the delay function you can simply use Step Out function. It will execute the delay function and take execution to the next line immediately after the delay function.

Figure 4.47

12. Now press the Reset button to Reset the CPU and observe the PLL window.

Figure 4.48

13. Observe that CCLK is also reset and now it is equal to XTAL frequency. Similarly the GPIO and pin connect block vales are also reset.

Figure 4.49

14. Now let us setup a break point at line no. 146 i.e. at the beginning of the main function. It will halt the execution when the debugger encounters this line.

Figure 4.50

15. Now click Run to start the execution.

Figure 4.51

16. After clicking on RUN observe the change in the CCLK field. The execution has also halted at the breakpoint that was setup earlier. Now again if you click Run the code will continue to execute.

Figure 4.52

After debugging is done select “Stop Debugging” from Debug menu. Alternatively you can press Ctrl+F5 to Start/Stop debugging.

4.7 Loading Hex file on the microcontroller using on-chip boot loader

LPC2148 microcontroller incorporates a capability to self program itself using the on-chip UART boot loader. This eliminates the need of an external programming hardware. The boot loader code is dedicated to use UART0 on the microcontroller. Once the code is loaded on the microcontroller UART0 is free to be used in the application. The Flash Magic software is used on the computer side for loading the hex file. The installation procedure is already explained in section 4.2. Before loading the hex file it is essential to setup Flash Magic for LPC2148 microcontroller. The microcontroller can be entered into the boot loader mode by pressing the Boot switch and then pressing Reset switch. This process is explained in detail in later sections.

1. Click **Start>Programs>Flash Magic>Flash Magic** to start Flash Magic utility.
2. Click **Select Device** to select LPC2148 microcontroller.

Figure 4.53

3. Expand the ARM7 tree and select **LPC2148** from the displayed list of microcontrollers.

Figure 4.54

4. After selecting the microcontroller, select the COM port that you are going to use and ensure that other settings are as shown in the figure below.

Figure 4.55

5. To select the hex file click **Browse** and point to the folder that was created earlier and select the hex file.

Figure 4.56

6. Connect a DB9 cable between the selected COM port on your PC and robot's serial port and turn ON the robot.

7. Press the Boot switch and while keeping it pressed, press Reset switch once. This will make LPC2148 enter into the boot loader mode

Figure 4.57

8. After ensuring the hardware setup in step 7 click start to begin programming the hex file.

Figure 4.58

9. Observe the status bar at the bottom of the Flash Magic tool. Wait for the "**Finished"** status and press "**Reset"** on the LPC2148 development board and observe the LEDs.

Figure 4.59

If not successful then repeat again from step 7.

4.8 Configuring Startup.s File using Configuration wizard

The "**startup.s**" is the collection of assembly instructions that is supposed to be executed at the start-up of the application code. The startup file contains information about the PLL settings, peripheral clock divider, stack configuration, etc. This file can be found in the Keil installation directory. On creating a new project keil allows you to copy this file to the project source folder so that you can modify the settings as per your application requirements without modifying the original file. To execute the start up code on reset please ensure that following setting is done in the target options window. To open this window goto Projects-> Options for Target 'Target1' from the menu bar.

Figure 4.60

1. In the Project explorer double click on **"startup.s"** file.

Figure 4.61

2. The startup.s file will open up as assembly file. To open the wizard click **Configuration Wizard** as shown in figure below.

Figure 4.62

3. Expand the PLL Setup tree to observe the default the PLL settings.

Figure 4.63

To include the PLL code in the startup file select the checkbox next to PLL setup heading. Here MSEL setting indicates that the onboard crystal frequency (12MHz) will be multiplied by 5. This indicates that the CPU core frequency will be 60MHz and this is the maximum operating frequency for LPC2148 microcontroller. Care should be taken while selecting the multiplier value to ensure that the CPU core frequency is not beyond the maximum value. When using the PLL setup the minimum onboard crystal frequency should be at least 10 MHz.

The PLL divider selection maintains the frequency of internal current controlled oscillator in the range of 150-320MHz. For more information please refer LPC2148 user manual. The default setting for **MSEL** is 5 and for **PSEL** it is 2. So the core frequency (FCLK) is 60MHz.
To change any setting just click on it and make use of the up/down arrows.

Figure 4.64

4. Now expand VPBDIV Setup tree and observe its settings. It is used to prescale the peripheral clock (PCLK).

Figure 4.65

Observe that in the default mode it is not included but at reset the VPBDIV loads the default value that is ¼ of the CPU clock. Using the configuration wizard you can set the peripheral clock as:
¼ of CPU clock or
½ of CPU clock or
equal to CPU clock.

For other settings please refer LPC2148 microcontroller user manual.
After you have done using configuration wizard rebuild the project.

5. Pin Functionality

LPC2148 microcontroller pin configuration:

Pin no.	Pin name	USED FOR
1	P0.21/PWM5/AD1.6/CAP1.3	PWM for right motor.
2	P0.22/AD1.7/CAP0.0/MAT0.0	Logic input 1 for Left motor (Left back)
3	RTXC1	RTC Crystal(32.768KHz) pin 1
4	P1.19/TRACEPKT3	LCD control line RS (Register Select)
5	RTXC2	RTC Crystal(32.768KHz) pin 2
6	VSS	**Ground:** 0 V reference
7	VDDA	Analog 3.3 V Power Supply
8	P1.18/TRACEPKT2	LCD control line RW(Read/Write Select)
9	P0.25/AD0.4/Aout	Buzzer
10	D+	USB bidirectional D+ line.
11	D-	USB bidirectional D- line.
12	P1.17/TRACEPKT1	LCD control line EN(Enable Signal)
13	P0.28/AD0.1/CAP0.2/MAT0.2	ADC input for white line sensor 2
14	P0.29/AD0.2/CAP0.3/MAT0.3	ADC input for white line sensor 3
15	P0.30/AD0.3/EINT3/CAP0.0	TSOP 1738 Input Capture pin via jumper J3. GPIO pin available on expansion slot.*
16	P1.16/TRACEPKT0	Sharp IR ranges sensor 2, 3, 4 and red LEDs of white line sensor 1, 2, 3 enable/disable pin** Turns off these sensors when output is logic 1
17	P0.31	USB Good Link LED indicator. It is ON during USB communication, otherwise it is OFF.
18	VSS	**Ground:** 0 V reference
19	P0.0/TXD0/PWM1	Transmitter output for UART0/PWM for servo motor***
20	P1.31/TRST	TSOP1738 Input Pin via jumper J2 and J3****
21	P0.1/RxD0/PWM3/EINT0	Receiver input for UART0
22	P0.2/SCL0/CAP0.0	I2C bus / GPIOs/Input Capture (Available on expansion slot of the microcontroller socket)
23	VDD	3.3 V Power Supply
24	P1.26/RTCK	GPIO/JTAG Pin(Available on expansion slot of the microcontroller socket)
25	VSS	**Ground:** 0 V reference
26	P0.3/SDA0/MAT0.0/EINT1	I2C bus / GPIOs/Input Capture (Available on expansion slot of the microcontroller socket)
27	P0.4/SCK0/CAP0.1/AD0.6	ADC input for Sharp IR range sensor 2 or IR proximity sensor 2 via 0E jumper*****
28	P1.25/EXTIN0	LCD data line(DB7)
29	P0.5/MISO0/MAT0.1/AD0.7	ADC input for Sharp IR range sensor 4 or IR proximity sensor 4 via 0E jumper*****
30	P0.6/MOSI0/CAP0.2/AD1.0	ADC input for Sharp IR range sensor 3
31	P0.7/SSEL0/PWM2/EINT2	PWM for left motor.

32	P1.24/TRACECLK	LCD data line(DB6)
33	P0.8/TXD1/PWM4/AD1.1	UART 1 transmit for XBee wireless module (if installed)
34	P0.9/RxD1/PWM6/EINT3	UART 1 receive for XBee wireless module (if installed)
35	P0.10/RTS1/CAP1.0/AD1.2	Logic input 2 for Right motor (Right forward)
36	P1.23/PIPESTAT2	LCD data line(DB5)
37	P0.11/CTS1/CAP1.1/SCL1	Logic input 1 for Right motor (Right back)
38	P0.12/DSR1/MAT1.0/AD1.3	ADC input for white line sensor 1
39	P0.13/DTR1/MAT1.1/AD1.4	ADC input for battery voltage monitoring
40	P1.22/PIPESTAT1	LCD data line(DB4)
41	P0.14/DCD1/EINT1/SDA1	Boot loader entry switch. External interrupt for on board boot switch in application mode.******
42	VSS	**Ground:** 0 V reference
43	VDD	3.3 V Power Supply
44	P1.21/PIPESTAT0	Logic input 1 for Left motor (Left back)
45	P0.15/RI1/EINT2/AD1.5	External Interrupt for the left motor's position encoder
46	P0.16/EINT0/MAT0.2/CAP0.2	External Interrupt for the right motor's position encoder
47	P0.17/CAP1.2/SCK1/MAT1.2	(Available on expansion slot of the microcontroller socket)
48	P1.20/TRACESYNC	GPIO/JTAG Pin(Available on expansion slot of the microcontroller socket)
49	VBAT	RTC Power Supply from 3V cell or 3.3V power supply.
50	VSS	**Ground:** 0 V reference
51	VDD	3.3 V Power Supply
52	P1.30/TMS	GPIO/JTAG Pin(Available on expansion slot of the microcontroller socket)
53	P0.18/CAP1.3/MISO1/MAT1.3	(Available on expansion slot of the microcontroller socket)
54	P0.19/MAT1.2/MOSI1/CAP1.2	(Available on expansion slot of the microcontroller socket)
55	P0.20/MAT1.3/SSEL1/EINT3	Not Connected
56	P1.29/TCK	GPIO/JTAG Pin(Available on expansion slot of the microcontroller socket)
57	RESET	Microcontroller Reset pin connected to onboard Reset Switch
58	P0.23	Indicates the presence of USB bus power
59	VSSA	**Analog Ground:** 0 V reference.
60	P1.28/TDI	GPIO/JTAG Pin(Available on expansion slot of the microcontroller socket)
61	XTAL2	Crystal 12.0 MHz
62	XTAL1	
63	VREF	ADC Reference Voltage(3.3V)
64	P1.27/TDO	GPIO/JTAG Pin(Available on expansion slot of the microcontroller socket)

Table 5.1: LPC2148 microcontroller pin connections

Note:

***** Pin 15 is tied to TSOP1738 output pin by shorting pads of jumper J3 on the microcontroller board. This pin is used to capture pulses generated by TSOP1738.

****** Sensor's switching can be controlled only if jumper J2 on main board is kept open.

******* Disconnect servo motor during ISP (In System Programming) or UART0 serial communication.

******** This pin is tied to TSOP1738 output pin via Jumper J2 and J3 on the microcontroller board. It is used to sense positive or negative edge of pulses coming from TSOP1738.

********* IR Proximity sensor no. 2 & 4 are connected to AD0.6 and AD0.7 via 0E resistors. These pins are directly connected to Sharp IR range finder no. 2 & 4. So to use Sharp IR range finder 2 & 3 instead of IR proximity 2 & 4 desolder these resistors from the main board. Refer Section 8.7.11 to know more about IR Proximity Sensors.

********** In application mode switch connected to this pin can be used to interrupt microcontroller. Otherwise it is used as entry mechanism for boot loader.

6. PC Based Control Using Serial Communication

Fire Bird V ARM7 LPC2148 robot can be controlled through the serial port of the host PC in wired or wireless mode. In order to do this you have to load appropriate hex file (firmware) on the robot. Refer to chapter 4 for how to load hex file.

The communication between the host computer and Fire Bird V robot is made by sending and receiving commands and data byte by byte. The host computer interacts with the Fire Bird V robot by first sending a command byte followed by data byte or it waits for data from the robot. All communication is initiated by host PC. The host computer acts as a master and Fire Bird V robot acts as a slave. PC introduces delay of at least 3ms between two consecutive commands, so that the microcontroller gets sufficient time to process the command sent to it.

You can also use NEX Robotics USB to Serial converter to control robot over USB or use NEX Robotics ZigBee wireless communication modules to control robot over wireless data link. GUI based robot control is covered in Chapter 7.

6.1 Communication Protocol

Communication protocol used for robot control is given below. You can build your own application by using command set given below to control robot using serial, USB or over wireless communication. It's a simple byte based protocol in which upper nibble is command and lower nibble can be data or command.

Commands to set velocity of the left and right motor:

The motor's velocity can be varied by writing the proper byte into the particular register, which generates a pulse width modulation (PWM) signal with 8 bit resolution. The value of the velocity control register can be set between 00 to FF hex, where a value 0 indicates that the motor is stopped and 0xFF indicates motor is running at full speed.

Command (HEX)	Function
1	Load the lower nibble of the left motor velocity control byte into the robot.
2	Load the upper nibble of the left motor velocity control byte into the robot and execute the command.
3	Load the lower nibble of the right motor velocity control byte into the robot.
4	Load the upper nibble of the right motor velocity control byte into the robot and execute the command.

Table 6.1

Example: Set left motor's speed control byte to 0xAB

To set the speed of the left motor to 0xAB, follow the sequence of commands below. Attach lower nibble "B" with command 1 and upper nibble A with command 2.

Step1: *Send 0x1B* *Load the lower nibble of the left motor speed in the robot.*

Step2: *Delay of at least 3 milliseconds*
Step3: *Send 0x2A* *Load the upper nibble of the left motor speed in the robot and execute the command*
Step4: *Delay of at least 3 milliseconds before loading next command*

Note: It is very important that you send the byte containing command 1 first and then send the byte containing command 2 for proper operation. The same rule is applicable for commands 3 and 4.

Commands to set direction of the robot:

ROBOT DIRECTION	**LEFT BWD (LB) P0.22**	**LEFT FWD(LF) P1.21**	**RIGHT FWD(RF) P0.10**	**RIGHT BWD(RB) P0.11**	**PWM P0.7 for left motor P0.21 for right motor**
FORWARD	0	1	1	0	As per velocity requirement
REVERSE	1	0	0	1	As per velocity requirement
RIGHT *(Left wheel forward, Right wheel backward)*	0	1	0	1	As per velocity requirement
LEFT*(Left wheel backward, Right wheel forward,)*	1	0	1	0	As per velocity requirement
SOFT RIGHT*(Left wheel forward,, Right wheel stop)*	0	1	0	0	As per velocity requirement
SOFT LEFT*(Left wheel stop, Right wheel forward,)*	0	0	1	0	As per velocity requirement
SOFT RIGHT 2 *(Left wheel stop, Right wheel backward)*	0	0	0	1	As per velocity requirement
SOFT LEFT 2 *(Left wheel backward, Right wheel stop)*	1	0	0	0	As per velocity requirement
HARD STOP	0	0	0	0	As per velocity requirement
SOFT STOP (Free running stop)	X	X	X	X	0

Table 6.2

Example: To set left motor velocity to 0x84, right motor velocity to 0x65, and move backward.

Step1: *0x14* *Load the lower nibble '4' of the left motor speed into the robot*
Step2: *Delay of at least 3 milliseconds*
Step3: *0x28* *Load the upper nibble '8' of the left motor speed into the robot and execute the command*
Step4: *Delay of at least 3 milliseconds*
Step5: *0x35* *Load the lower nibble '5' of the right motor speed into the robot*
Step6: *Delay of at least 3 milliseconds*
Step7: *0x46* *Load the upper nibble '6' of the right motor speed into the robot and execute the command*
Step8: *Delay of at least 3 milliseconds*
Step9: *0x52* *move backward*
Step10: *Delay of at least 3 milliseconds before loading next command*

Commands to access the Analog sensor data:

Command (HEX)	Data
60	**Battery voltage** Robot sends back 8 bit battery voltage value. To convert this value in to volts use the following conversion formula: Battery Voltage = ADC data x 0.069
62	**Front Sharp IR range sensor data (Front distance)** The Robot will return 8 bit value which indicates distance between the obstacle and front Sharp sensor.
63	**Sharp IR range sensor 2 data** The Robot will return 8 bit value which indicates distance between the obstacle and Sharp sensor.
64	**White line sensor 1 (Left)** The Robot will return an 8 bit analog value of the left white line sensor
65	**White line sensor 2 (Center)** The Robot will return an 8 bit analog value of the center white line sensor
66	**White line sensor 3 (Right)** The Robot will return an 8 bit analog value of the right white line sensor
67	**Sharp IR range sensor 4 data** The Robot will return 8 bit value which indicates distance between the obstacle and Sharp sensor.
C2	**IR Proximity sensor 2** The Robot will return an 8 bit analog value of the IR Proximity sensor 2

C4	**IR Proximity sensor 4** The Robot will return an 8 bit analog value of the IR Proximity sensor 4

Table 6.3

Commands to turn on / off the buzzer:

69	Turn on the buzzer.
6A	Turn off the buzzer.

Table 6.4

Robot Version Signature

6B	If 6B is sent to the robot will send back its ID

Table 6.5

Position encoder data:

Position encoder pulse count for the position tracking:

72	The robot will return lower byte of the pulse count for the left motor**.**
73	The robot will return upper byte of the pulse count for left motor.
79	The robot will return lower byte of the pulse count for the right motor**.**
7A	The robot will return upper byte of the pulse count for right motor.

Table 6.6

Note: To get an actual pulse count, combine the lower byte and upper byte to get a 16 bit value. For more information on the position encoder resolution refer to the section 3.7.

6.1B: Communication protocol for basic robot control

In the section 6.1, advanced protocol for complete robot control over GUI is covered. In this section a basic protocol for robot control is explained. Using this protocol you can move robot in all direction and turn on / off its buzzer.

To use this protocol you need to load following hex files on the robot which are located in the "Experiments" folder in the documentation CD depending on the type of communication mode used.
In all cases 9600 bps baud rate is used. Robot control covered in the section 6.2, 6.3 and 6.5 uses communication protocol mentioned in this section. Keys are mapped in the intuitive way on the Numerical pad of the keyboard.

Serial port: Serial_Comm.hex
XBee wireless module: Serial_Comm_Xbee.hex

Character	ASCII value	Action
8	0x38	Forward
2	0x32	Backward
4	0x34	Left
6	0x36	Right
5	0x35	Stop
7	0x37	Buzzer On
9	0x39	Buzzer Off

Table 6.7

Important:
While using "Numerical Pad" of the key board, make sure that "Num. Lock" is on.

6.2 Using the Terminal software to control robot via Serial Communication

Terminal is easy to use free software for serial communication. It is located in the "Software and Drivers" folder in the documentation CD. It can also be downloaded from http://hw-server.com/software/termv19b.html. Serial communication protocol covered in the section 6.1 can be used with the terminal software control robot over wire or wireless medium.

Connect serial cable between the robot and the computer as shown in figure 6.1. Load the "Serial_Comm.hex" file which is located in the "Experiments" folder in the documentation CD on the robot. For loading hex file onto the robot refer to chapter 4.

Figure 6.1

Step 1: Copy Terminal software on the PC from the "Software and Drivers" folder from the documentation CD and double click on the terminal software. The terminal window will open up.

Figure 6.2

Step 2: Select the COM Port and set the baud rate and other settings as desired. Once all the settings are done click connect. If you are using serial port then select COM1. If you are using NEX Robotics USB to serial converter or Wireless ZigBee module then refer to section 6.3 and 6.4 for how to identify COM port. Set the baud rate at 9600 bps

Figure 6.3

Step 3: Now you are ready to transmit the data. Type the data into the text box and click send. For more information about using terminal software click help.

Figure 6.4

Step 4: For controlling robot motion and buzzer use "Numerical Pad" Use commands from the table 6.7 to control the robot. Make sure that "Num lock" is on for the numeric keypad.

Figure 6.5

6.3 Robot control using USB port

Fire Bird V ARM7 LPC2148 can be controlled through computer's USB port by simply using a USB to serial converter designed by Nex Robotics. This communication requires "Serial_Comm" to be loaded on the robot which is located in the "Experiments" folder in the documentation CD.

For using Fire Bird V in USB Communication mode place connect DB9 cable between USB to serial connector and robots serial port as shown in the figure 6.6 and connect the USB cable between PC and USB to serial converter.

Figure 6.6: Jumper setting to enable USB communication

Before using USB port we need to install the driver software for FT232 USB to serial converter. The software is located in the "Software and Drivers \ CDM 2.06.00 WHQL Certified" folder provided in the Fire Bird V CD or can also be downloaded from the NEX Robotics' website.

After installation is complete follow the step 1 to 4 of the section 6.2 to control the robot using keyboard by using the "Terminal" software.

Steps to install the drivers for USB to serial converter:

Step 1:
Copy the driver installation folder on your PC from "Software and Drivers \ CDM 2.06.00 WHQL Certified" Folder in the CD.

Step 2:
Connect the USB to serial converter cable between robot and the PC

Step 3:
On connecting the device "Found New Hardware" message will appear in the taskbar tray and the following window opens.

Figure 6.7

Step 4:
Check on the radio button "No, not this time" and then click on the next button.

Figure 6.8

The following window will appear.

Figure 6.9

Select the second option manually to install the drivers and click on next button.

Step 5:
Now check the second option and set the location of folder containing drivers E.g.(C:\CDM 2.06.00 WHQL Certified).

Figure 6.10

Step 6:
On clicking next driver installation will begin.

Figure 6.11

Step 7:
On successfully installing the driver following window will appear. Click Finish to complete the installation.

Figure 6.12

After installation of FT232 USB UART software, PC may ask for USB serial port software. To install this software follow steps 1 to 7 of USB serial converter software installation.

6.4 Identifying COM Port number of the USB to serial converter or ZigBee wireless module

To use terminal.exe or any other serial program for robot control we need to first identify communication port which is generally referred as COM n, i.e. COM1 or COM2 etc. on which USB to serial converter or wireless device is connected. Follow these steps to identify your COM Port number.

Step 1:
Right Click My Computer and click on properties. System properties window will appear.

Figure 6.13

Step 2:
Click on the Device manager in the Hardware tab.

Figure 6.14

Step 3:
Expand Ports (Com & LPT) tree. COM Port number is mentioned in the parenthesis next to USB Serial Port.

Figure 6.15

Step 4:
If the COM port number is greater than 10 Terminal will not be able to detect it. To resolve this problem, change the port number by right clicking on “USB serial Port” and select properties.

Figure 6.16

In the Port settings tab click on the **Advanced…** button, the following window will appear.

Figure 6.17

You can change the COM port number by clicking on the Com Port number drop down list and select the appropriate number. Make sure the new COM port is not being used by any other device.

6.5 Robot control using ZigBee wireless communication module

Figure 6.18: ZigBee Wireless USB Module from NEX Robotics

To control robot via wireless communication, load “GUI_Control_Xbee.hex” on the robot which is located in the “GUI and Related Firmware” folder.

USB Wireless Module from the NEX Robotics enables wireless transmission of serial data through PC’s USB port. It uses Xbee module for wireless communication. The Xbee module can be configured via PC’s USB port easily using X-CTU utility to change frequency, baud rate etc. It is located in the “Software and Drivers” folder in the documentation CD. On the PC side this device is treated as the Communication Device Class (CDC) of USB family and it is assigned a virtual Comport number allowing the user to make use of existing GUI. The destination side (robot) requires another Xbee wireless module.

Make sure that Xbee wireless module s installed on the robot. Refer to chapter 9 for how to install Xbee wireless module on the robot.

We need to load “Serial_Comm_Xbee.hex” on the robot which is located in the “Experiments” folder in the documentation CD.

After all the driver installation is complete follow the step 1 to 4 of the section 6.2 to control the robot using keyboard using the “Terminal” software.

Driver Installation

Step 1:
Locate the wireless communication folder from the CD

Step 2:
Copy “Software and Drivers \ CDM 2.06.00 WHQL Certified” folder from the installation CD.

Step 3:
Follow the same process as in section 6.3 to install the driver.

Step 4:
Install ZigBee wireless module on the main board as shown in figure 6.19. Refer to chapter 9 for more details.

Step 5:
Identify Com port for the communication on the PC. Turn on the robot and start wireless communication.

Note: Wait for at least 5 seconds to start the wireless communication after turning on the robot and the USB wireless module.

Figure 6.19: ZigBee module on the main board

7. Robot Control using 'GUI' for Fire Bird V ARM7 LPC2148

Fire Bird V ARM7 LPC2148 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 communicates with robot at the 115200 baud rate.

7.1 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/USB serial communication: "GUI_Control_Serial.hex"
XBee wireless module based communication: "GUI_Control_Xbee.hex"

The hex files are located in the "GUI and Related Firmware" folder in the CD.

For information on how to load hex file, refer to chapter 4.

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

7.2 Installing GUI

Step1: Copy “FIRE BIRD V ARM7 LPC2148” folder which is located inside the “GUI and Related Firmware” folder.

Inside this folder Click “setup.exe” to start the installation process. The GUI requires Microsoft.net framework to be installed on the PC. If .net framework is not found on PC it will prompt to install from the setup itself. If .net framework is already installed you may proceed to step 4.

Step 2: Click **Accept** to install .net framework.

Figure 7.1

Step 3: The .net framework files will be copied to your system. Please wait until it is complete.

Figure 7.2

Step 4: After installing .net framework setup will start installing GUI software. The FBV ARM7 LPC2148 setup welcome screen will appear. Click **Next** to start installation.

Figure 7.3

Step 5: Select the folder where GUI software will be installed and click **Next** to continue.

Figure 7.4

Step 6: Click **Next** to confirm the installation.

Figure 7.5

Step 7: Wait until installation is complete.

Figure 7.6

Step 8: Click **Close** to finish the installation.

Figure 7.7

7.3 Using GUI

Step 1: After successful installation go to Start -> All Programs -> FIRE BIRD V ARM7 LPC2148 -> FIRE BIRD V ARM7 LPC2148 or click on Fire Bird V ARM7 LPC2148 icon on desktop location to open GUI.

Figure 7.8

Step 2: Connect Robot with the PC using serial cable / NEX Robotics USB to serial converter or with the ZigBee wireless module.

Step 3: Select the appropriate COM port. If USB to serial converter module from NEX Robotics or USB ZigBee wireless module is used then identify COM port as mentioned in the section 6.4. Select the correct COM port number and click on connect.

The robot is ready to be controlled using GUI

Figure 7.9: GUI showing robot's data

7.4 Robot control using XBee wireless module

To control robot using ZigBee wireless module refer to section 6.5. All the process remains the same as mentioned in the section 7.1, 7.2 and 7.3. Only difference is that instead of USB to serial converter, NEX Robotics wireless ZigBee USB module needs to be connected to the PC.

8. Hardware Description

This chapter contains the hardware description of the Fire Bird V ARM7 LPC2148 robot. In this chapter full expansion capability of the robot is covered so that user can do custom modifications as per the requirements.

Fire Bird V platform consists of 6 modules:

1. Power management
2. Sensing
3. Actuation (locomotion)
4. Other peripherals
5. Communication
6. Intelligence (microcontroller)

Figure 8.1: Fire Bird V ARM7 LPC2148 robot top view

Figure 8.2: Fire Bird V ARM7 LPC2148 bottom view

8.1 Connections

Figure 8.3: ARM7 LPC2148 microcontroller adaptor board for the Fire Bird V robot

Figure 8.4: Main board

Figure 8.5: Bottom view of the main board

8.1.1 Socket connections for the microcontroller adaptor board on the main board

Figure 8.6: Microcontroller adaptor board socket connections on the main board

8.2 Power management module on the main board

Power management block on the Fire Bird V performs following functions.

1. Battery voltage monitoring
2. Battery current sensing*
3. Battery low warning in case battery is below critical level
4. Regulated supply for onboard payload.

* Current sensing is an optional accessory.

8.2.1 Battery

Fire Bird V is powered by 9.6V, 2.1Ah rechargeable Nickel Metal Hydride battery pack. When fully charged, battery pack gives 11.5V and when it is fully discharged, voltage drops to about 8.5V. Battery has built-in thermistor for monitoring battery temperature during battery charging.

Figure 8.7: 9.6V, 2000mAH rechargeable NiMH battery and pin connection

⚠ The NiMH battery provided with Fire Bird V has a connector which will fit into the above shown connector in only one orientation. Do not force the connection in any other way.

Figure 8.8: Battery connections for Fire Bird V ARM7 LPC2148

Pin Number	Function
1,2,9,10	Battery Positive
3,8	Thermistor Connection
4,5,6,7	Battery Negative

Table 8.1: Battery connections

8.2.2 Power sources and voltage regulation on the main board

Although Fire Bird V can be powered by NiMH battery but in order to do experiments for longer duration without worrying about the battery getting low, robot can be powered by external power source which is also called as auxiliary power source. Auxiliary power source

provides regulated 12V, 1A supply to the robot. You can apply up to 18V to the robot without any problems. Beyond 18V voltage regulators on the robot will start failing and system will get damaged. The entire supply lines combined together robot can supply maximum of 1Ampere current to the external load when operation on the battery.

Most of the systems on board the robot require 5V regulated supply. Only wireless module requires 3.3V supply. Robot's main board has three voltage regulators and two additional power lines for different types of functionalities.

Depending on the power switch position and the presence of the auxiliary power robot can run either on the battery or on the auxiliary power source. Supply is further divided in two types.

1. V Batt Supply

"V Batt Supply" stands for stabilized supply coming from the battery. When battery is almost discharged (about 30% power remaining) and onboard payload draws large current in excess of 2 amperes then battery voltage can fall below 6.3V momentary. Voltage regulators will not be able to function properly and its output will fall below 5V if input voltage falls below 6.3V. If output voltage falls below 5V then microcontroller can get reset. To extend the usable battery life and to reduce the probability of microcontroller getting reset diodes D7 and D8 along with the capacitor C54 is used. When battery voltage suddenly drops, diode D7 and D8 prevents the reverse flow of the current and capacitor C54 maintains voltage with in safe limits for about 300 milliseconds. For this duration capacitor C54 acts as small battery. Similar arrangement is done in the "V Mot Supply" using diodes D9, D10 and capacitor C53. This scheme extends usable range of the fully charged battery.

2. V Mot Supply

"V Mot Supply" stands for motor supply. It is used to power DC motors and other heavy loads having lots of current fluctuations. It is the nosiest supply line on the robot. It should be used for heavy loads which requires large amount of current. This supply can varied between 8V to 11.7V depending on the battery state and type of power source (battery / auxiliary power) used. This line can supply additional 800mA to the external load.

Figure 8.9: Voltage regulators on the main board

3. 5V System

"5V System" is used to power various modules of the robots which does not require high current and where voltage stability is very important. It is used to power logic

supply of the ICs, sensing elements of the sensors, LCD etc. It is the most stable source of the supply on the main board. It can source 400mA current for the external load.

4. 5V Sensor

"5V Sensor" is used to power the active elements of the sensors and sensor modules. Its voltage can fluctuate by about 100mV if sensors are turned on or off. In fully loaded Fire Bird V robot this supply should not be used to power external load having current requirement more than 100mA.

Figure 8.10: On / off switch position

8.2.3 Current sensing

Fire Bird V robot can sense its current consumption using Hall Effect current sensor ACS712.

Figure 8.11: Current sensing in Fire Bird V

ACS712's current sensing element is located between robot's ground and the battery ground. When no current is flowing through the sensor, it gives 2.5V output. This output value reduces by 185mV / ampere of current flow if 5 Ampere type sensor is installed. If 20

Ampere type sensor is installed value is reduced by 100mV / ampere. This sensor is an optional accessory. When this sensor is absent its sensing path is shorted with 0 ohm resistor.

8.2.4 Battery low indication

Figure 8.12: Battery low indication

Fire Bird V robot uses 8 cell NiMH battery. Battery is considered as fully discharged if its voltage drops below 8V. If robot is operated while battery is low, its performance will be inconsistent. To prevent any abnormality in the robot's behavior battery low indication is used. When robot's battery voltage goes below 8V robot starts giving beeping sound.

Zener diode D1 produces stable 5.1V reference irrespective of change in the battery voltage. This voltage is buffered by unity gain opamp module IC7B. Threshold value for the battery low warning is set by the potentiometer R92. Its value is buffered using module IC7A. Reference voltage and voltage form the potentiometer R92 are compared and difference is amplified with the gain of 45.45 (100K / 2.2K) in the module IC7D. Output coming from the module IC7D is buffered with IC7C unity gain opamp and given to the transistor Q5 which drives the battery low warning buzzer.

Figure 8.13: Various components of the power management module

8.3 Sensors

Fire Bird V robot contains many types of sensors on the main board, microcontroller adaptor module and on the separate boards. We will cover sensors located on the microcontroller board in the separate section.

If a sensor emits some signal and senses any object based on the reflected signal then it is known as active sensor. If many robots in the same field are using active sensors such as Sharp IR range sensors, IR proximity sensors or ultrasound sensors then they will interfere with the sensors of the other robots.

To prevent this interference from occurring Fire Bird V can switch all of its active sensors on and off. This means that many Fire Bird series robots can work in same field without causing interference with other Fire Bird robot's sensor. Many robots can synchronize when particular sensor should be turned on or off over wireless communication using ZigBee wireless modules. Another advantage of this facility is that robot can turn off these sensors when not in use to conserve power.

8.3.1 Sharp IR range sensors

For accurate distance measurement, the robot uses Infrared Range sensors. Up to 5 Sharp IR range sensors can be installed on the Fire Bird V main board. However, LPC2148 robot currently supports only three such sensors. For more details refer comparison chart. Fire Bird V supports three types of IR range sensors from Sharp Microelectronics.

GP2D120 GP2D12 GP2Y0A02YK

Figure 8.14: Sharp IR Range sensors for Fire Bird V

Sensor	Range	Blind Spot
GP2D120X	30cm to 20cm	4cm to 0cm
GP2D12	80cm to 10cm	10cm to 0cm
GP2Y0A02	150cm to 20cm	20cm to 0cm

Table 8.2: Sharp IR Range sensors coverage

Figure 8.15: Sharp Sensors mounted on Fire Bird V

Working of the Sharp IR range sensors is covered in detail in the section 3.8 of the chapter 3. Sharp IR range sensor is an active sensor i.e. it emits signal (IR beam) and measures angle of reflection for distance from the obstacle.

In order to prevent interference with other robot's sensors all the sharp sensors can be turned on or off by the microcontroller.

Figure 8.16: Sharp IR range sensors

Sharp IR range sensor 1 and 5 are controlled by the MOSFET Q4. Sharp IR range sensors 2, 3 and 4 are controlled by the MOSFET Q2. These sensors can be permanently enabled by putting jumper J3 and J2 on the main board. In the normal case when input is less than 0.65V or there is no input at the Pin 35 and 40 of the main board's socket, corresponding transistors T16 and T9 are off. Gate of the MOSFETs Q4 and Q2 gets 5V which drives them in to saturation and sensors are turned on. When input voltage more than 1V is applied at the Pin 35 and 40 of the main board's socket, corresponding transistors T16 and T9 goes in to saturation, there by grounding gates of the MOSFETs Q4 and Q2 which turns off the MOSFETs and sensors are switched off.

Sensors	Main board socket pin number	Logic state	Jumper on the main board to turn on sensors permanently
Sharp IR range sensor 1,5	34	0: Sensors are turned on 1: Sensors are turned off	J3
Sharp IR range sensor 2,3,4 and LEDs of the white line sensors (left, center, right)	40	0: Sensors are turned on 1: Sensors are turned off	J2
IR proximity sensors 1 to 8	33	0: Sensors are turned on 1: Sensors are turned off	J4

Table 8.3: Sensor power control pins and jumpers

Note:

- Sharp IR Range sensors 1 and 5 are not available on LPC2148 robot and so its pin no.34 on main board is a not connected pin.
- In LPC2148 robot IR proximity sensors cannot be enabled/disabled under software control. LPC2148 robot supports only IR proximity sensor 2 and 4. To use this sensor insert jumper J4 permanently.

Fig 8.17: Infrared Range finder sensor and its inside view

Figure 8.18: Installing Sharp IR range sensor on the main board

Note:

- MOSFET Q2 also controls the turning on / off of the red LEDs of the white line sensors.
- If you are not going to switch sensors on / off using MOSFETs then corresponding jumpers as per the table 8.4 should be shorted on to turn on sensors permanently.
- Sharp GP2Y0A02 sensors body is made up of conductive plastic. Hence foam tape is added as an insulator between the sensor and the metal strip. If this isolation is not provided then sensor will get direct ground path and switching MOSFET will get bypassed which is not recommended.

8.3.2 IR Proximity and directional light intensity sensors

Infrared proximity sensors are used to detect proximity of any obstacles near the robot. Directional light intensity sensor gives intensity of the light in particular direction. IR proximity sensor consists of IR LED and IR photo diode. When obstacle comes within sensing range IR photo diode detects reflected light of the IR LED. To reduce interference with ambient light photo diode is cased in material which allows only IR light to pass. In order to eliminate interference due to ambient light while sensing obstacle, reading should be taken while IR LED is on and off. By comparing these two values effect of ambient light can

be nullified. To detect directional light intensity, IR LED is turned off and light intensity at photo diode is measured. Fire Bird V has eight IR proximity sensors. IR LEDs of all these sensors can be turned on and off by MOSFET Q3. To turn on IR LEDs permanently J4 needs to be shorted. Refer to table 8.4 for more details. MOSFET operates in a same way as explained in section 8.3.1.

In the absence of the obstacle there is no reflected light hence no leakage current will flow through the photo diode and output voltage will be around 5V. As obstacle comes nearer, more light gets reflected and falls on the photo diode and leakage current flowing through the photo diode starts to increase and voltage across the diode starts to fall.

When IR proximity sensors used in conjunction with Sharp IR Range sensors, IR proximity sensors covers sensing range in which Sharp IR Range sensors have blind spot. IR proximity sensors working together with Sharp IR range sensors covers range from 0 to maximum range of the Sharp IR sensor without having any blind spot.

Figure 8.19: IR Proximity sensors

Figure 8.20: IR Proximity sensor positions

8.3.3 Sharp IR range sensors and IR proximity sensor status indicator and jumpers

Figure 8.22 shows jumpers and corresponding MOSFETs used to turn on / off the sensors. Figure 8.23 shows the LEDs which indicates the status of the sensors.

Figure 8.21: Sensor Control Switches and Jumpers

Name	Function
J1	LCD Backlight Enable/Disable
J2	Front Sharp and White line sensors Enable/Disable
J3	Side Sharp Sensors Enable/Disable
J4	IR Proximity(TCRT) Sensors Enable/Disable
M2	Front Sharp and White Line Sensors MOSFET
M3	IR Proximity (TCRT) MOSFET
M4	Side Sharp MOSFET

Table 8.4 Jumper functions

Figure 8.22: Sensor status indicator LEDs

8.3.4 White line sensors

White line sensor consists of a highly directional phototransistor for line sensing and red LED for illumination. Due to the directional nature of the phototransistor it does not get affected with ambient light unless it is very bright.

Figure 8.23: Schematic of white line sensor

White line sensor consists of ultra bright red LED and highly directional phototransistor. Ultra bright red LED illuminates the ground surface and the highly directional phototransistor senses the reflected light from very small surface. White line sensor can be calibrated for different types of operating conditions by adjusting the light intensity of the red LEDs. This can be done by tuning potentiometers R65 (left white line sensor), R57 (center white line sensor), R55 (right white line sensor). 100 ohms resistors which are in the series of the potentiometers R65, R57 and R55 protect red LEDs from the excess of current if potentiometers tuned to 0 ohms. Potentiometers are factory set for the best performance. There values should not be changed unless required. LEDs are powered by 5V sensor supply while photo transistors are powered by 5V system supply for better stability. Resistors R48, R23 and R28 determine the sensitivity of the phototransistor. By repeated experimentations we found 100K ohms as the optimal value for best performance in all the conditions.

Figure 8.24: Potentiometers for white line sensor calibration

Figure 8.25: White line sensor pin connections

Figure 8.26: White line extension to mount 5 or 7 channel white line sensor

Pin No.	Function
1,2	White line sensor 1 (Left sensor) Data Out
3,4	White line sensor 2 (Centre sensor) Data Out
5,6	White line sensor 3 (Right Sensor) Data Out
7,8	White line sensor 1 LED
9,10	White line sensor 2 LED
11,12	White line sensor 3 LED
13,14	System Supply, 5V
15,16	LED common cathode
17,18	GND
19,20	Sensor Supply, 5V
21	Expansion slot for White line sensor 5
22	Expansion slot for White line sensor 4
23	Expansion slot for White line sensor 6
24	Expansion slot for White line sensor 7

Table 8.5: White line sensor pin connections

In the default condition Fire Bird V is interface with the 3 channel white sensor. Robot can also be easily interfaced with the 5 or 7 channel white line sensors. In the 3 channel white line sensor 20 pin FRC connector is used. To make support for 5 or 7 channel white line sensors 26 pin FRC connector needs to be added. In the figure 8.23 and figure 8.26, pins Z2-1 to Z2-4 are used for connecting additional 4 white line sensors. Figure 8.26 shows the location of these pins on the main board. Refer comparison chart for detailed information on no. of sensors that can be attached to LPC2148 robot.

8.3.5 Other sensors

Fire Bird V robot have many more sensors such as current sensor, position encoders, TSOP1738 IR receiver, Voltage sensing and lots of add-on sensors which will be covered in the different topics in the manual.

8.4 Actuation (locomotion)

Fire Bird V robot has four motion control ports. Figure 8.30 shows the location of the required ports. Each Port can drive one DC motor and also provides connections for the position encoder. Motor is driven by L293D motor driver IC. Each IC can drive two DC motors with maximum current rating of 500 mA. Fire Bird V's main board has two L293D ICs.

"5V system supply" is used for driving L293Ds logic circuits. "V Batt Mot" is used to supply power for motor driving. C22 – C25 and C29 – C32 is used for noise suppression.

Optical encoder MOC7811 is used for position encoder on the robot. It consists of IR LED and the photo transistor mounted in front of each other separated by a slot in the casing. When IR light falls on the photo transistor it gets in to saturation and gives logic 0 as the output. In the absence of the IR light it gives logic 1 as output. A slotted encoder disc which is mounted on the wheel is placed in between the slot. When encoder disc rotates it cuts IR illumination alternately because of which photo transistor gives square pulse train as output. Output from the position encoder is cleaned using Schmitt trigger based inverter (not gate) IC CD40106.

Logic signals to drive the two L293D comes from the pins 47 to 55 and pins 66 to 68. Logic level on the pins 47 to 55 can be seen on the bar graph LED display on the main board. Figure 8.32 shows the connections of pins from the main board socket in general.

Figure 8.27: Schematic of the motion control module

If you want to drive bigger robot using Fire Bird V main board then L293D motor drivers on the main board cannot provide sufficient power. In such case we can use drive high power motor drivers such as Hercules 30 Amp. Mini Motor Driver (NR-MDR-02) from NEX Robotics for driving these motors. Figure 8.30 shows the location of the external motor interface port. All the logic signals coming from the pin 47 to 55 and ground pin is connected to this port. User can solder 20 pin FRC connector and interface these pins to the external motor drivers. Up to 3 Hercules 30 Amps. Mini Motor Driver (NR-MDR-02) can be interface with this port. Logic state on these pins can be observed on the bar graph LED display.

Figure 8.28: DC geared motors and position encoders

Figure 8.29: Position encoder assembly

Figure 8.30: Motion control connections on the main board

Connector Name	Description
L	Left motor connector 1
R	Right motor connector 2
C1	C1 motor connector
C2	C2 motor connector
External Motor Interface Port	Logic signals of pins 47 to 55 of the main board socket for interfacing external high current motor drivers

Table 8.6: Use of connectors of the motion control module

Pin No.	Motor and encoder module wire colors	Function
1	Black	VCC, 5V System
2	Brown	Position Encoder data
3	Red	NC
4	Orange	GND
5	Yellow	Motor 2
6	Green	Motor 1

Table 8.7: Motor drive port pin connections

Figure 8.31: L293D motor driver ICs on the main board

Figure 8.32: Motion status LED indication

Figure 8.33: Motion status LED indication for ARM7 LPC2148 microcontroller adaptor board for Fire Bird V

Figure 8.34: Hercules 30 Amp. Mini Motor Driver (NR-MDR-02) for driving high power motors

8.5 Other peripherals

8.5.1 LCD

Figure 8.35: LCD display schematics

Figure 8.36: LCD socket and other settings

LCD is interfaced to the pins 22 to 28 of the main board socket. LCD uses 5V System supply for its operation. For LCD backlight V Battery supply is used. Figure 8.36 shows LCD backlight jumper and LCD contrast control potentiometer. In order to save power LCD backlight can be turned off by removing LCD backlight jumper. LCD's contrast can be adjusted by LCD contrast control potentiometer.

Figure 8.37: LCD interfacing with the ARM7 LPC2148 microcontroller adaptor board

Microcontroller	LCD PINS	Description
VCC	VCC	Supply voltage (5V).
GND	GND	Ground
P2.4	E (Control line)	Enable
P2.5	-R/W (Control line)	-READ /WRITE
P2.6	RS (Control line)	Resistor select
P2.0 to P2.3	D4 to D7 (Data lines)	Bidirectional Data Bus
	LED+, LED-	Backlight control

Table 8.8: LCD pin mapping with the ARM7 LPC2148 microcontroller adaptor board

8.5.2 Buzzer

Figure 8.38: Buzzer

Buzzer is driven by BC548 transistor. Resistor R40 of 100K is used to keep transistor off if the input pin is floating. Buzzer will get turned on if input voltage is greater than 0.65V.

8.5.3 SPI expansion port on the main board

SPI expansion port will be used in future to add robotic arm, color sensing sensors etc. It has SPI communication pins and important types supply for powering these devices. Port can be seen in figure 8.5.

Figure 8.39: SPI expansion port pin functions

8.6 Communication

8.6.1 Serial communication

Serial port's pin 2, 3 and 5 are connected with the main board socket. Other pins of the serial port are also available on the main board. Figure 8.40 shows the serial port connections on the main board and table 8.9 gives the pin functions of the each pin.

Figure 8.40: Serial port pins

Figure 8.41: Serial port connections with the main board socket

Pin No.	Description	Comments
1	Data Carrier Detect (DCD)	Not Used
2	Receiver Data (RXD)	Used
3	Transmit Data (TXD)	Used
4	Data Terminal Ready (DTR)	Not Used
5	Signal Ground (GND)	Used
6	Data Set Ready (DSR)	Not Used
7	Request to Send (RTS)	Not Used
8	Clear to Send (CTS)	Not Used
9	Ring Indicator (RI)	Not Used

Table 8.9: Serial port pin out

8.6.2 USB communication

Fire Bird V's main board has USB port. All its pins are connected to the main board socket.

Figure 8.42: USB port connections with the main board socket

8.6.3 ZigBee wireless communication

Fire Bird V uses XBee wireless modules form Digi international (www.digi.com). LEDs W1, W2, W3 and W4 are used for status indication of the wireless module. Read the wireless module's datasheet for more details.

Figure 8.43: ZigBee wireless module schematics

Figure 8.44: ZigBee wireless module and LED indicators

LED	Connection to Xbee Wireless module Pin no.	Description
W1	13	On/Sleep Module Status
W2	15	Associate LED
W3	17	Reserved
W4	6	RX Signal Strength Indicator

Table 8.10: ZigBee LED Description

8.7 ARM7 LPC2148 microcontroller adaptor board

Figure 8.45: ARM7 LPC2148 microcontroller adaptor board

8.7.1 Power management

Power management block on the ARM7 LPC2148 microcontroller adapter board provides power to the microcontroller and other devices and the power for the servo motor.

ARM7 LPC2148 microcontroller adapter board has two low drop voltage regulators:

1. "3V regulator" supplies power to the microcontroller and its peripherals.
2. "5V regulator" supplies power to the servo motor.

Figure 8.46 Power Supply Circuit

8.7.2 Battery voltage sensing

Filtered battery voltage is used for battery voltage sensing. Analog to Digital Converter (ADC) can measure maximum voltage of 3.3V. Hence battery voltage is scaled down from 8-15V to less than 3.3V using resistor divider network formed by R12 and R13. It scales down the voltage by approximately 1/3 of the actual value.

LPC2148 ADC has 10 bit resolution. For battery voltage measurement only 8 bits are used for ADC conversion. To calculate voltage from the ADC's digital value we use following formula:

V Battery = 0.7V + (ADC value * (3.3V/255) * ((10K + 3.3K) / 3.3K))

V Battery = 0.7V + (ADC value * 0.05226)

In the above formula:

- 0.7V represents voltage drop across the diodes D7 and D8. Refer to figure 8.9 for more details.
- 3.3V/255 represents the ADC step resolution
- (10K + 3.3K) / 3.3K is a voltage divider formula

Figure 8.47 Battery Voltage Divider Bias Circuit

8.7.3 TSOP1738 RC5 IR Receiver and Decoder

TSOP1738 is an IR receiver and RC5 decoder. It is very commonly used in televisions for receiving commands from the remote control. It can be used to control robot using TV remote control or many robots can be controlled simultaneously if you make your own TV remote equivalent and interface it with the PC. Such type of setup can be used in the preliminary form of robo-soccer.

Figure 8.48 TSOP1738

8.7.4 LPC2148 Microcontroller Board Expansion Socket

LPC2148 is a feature rich microcontroller with lots of available I/O ports. Many of the ports are available on the uC Expansion Socket. Table 8.11 lists the connection details of all the pins of the socket.

Figure 8.49: Expansion Header on Microcontroller Board

Pin No.	Function
1	Reserved for future use
2	Reserved for future use
3	Reserved for future use
4	Reserved for future use
5	Reserved for future use
6	Reserved for future use
7	Reserved for future use
8	LPC2148 pin no. 47 (P0.17/CAP1.2/SCK1/MAT1.2) *2
9	LPC2148 pin no. 54 (P0.19/MAT1.2/MOSI1/CAP1.2) *2
10	LPC2148 pin no.48 (P1.20/TRACSYNC)
11	LPC2148 pin no. 53 (P0.18/CAP1.3/MISO1/MAT1.3) *2
12	LPC2148 pin no.34 (P0.9/RXD1/PWM6/EINT3)
13	LPC2148 pin no. 27 (P0.4/SCK0/CAP0.1/AD0.6)
14	LPC2148 pin no.3 (P0.8/TXD1/PWM4/AD1.1)
15	LPC2148 pin no.26 (P0.3/SDA0/MAT0.0/EINT1) *3
16	LPC2148 pin no. 22 (P0.2/SCL0/CAP0.0) *3
17	Not Connected
18	LPC2148 pin no. 29 (P0.5/MISO0/MAT0.1/AD0.7)
19	LPC2148 pin no. 19 (P0.0/TXD0/PWM1)
20	LPC2148 pin no. 21 (P0.1/RXD0/PWM3/EINT0)
21	GND
22	GND
23	GND
24	Not Connected
25	GND
26	Not Connected
27	GND
28	LPC2148 pin no. 57 (Reset)
29	GND
30	LPC2148 pin no. 64 (P1.27/TDO)(JTAG) *1
31	GND
32	LPC2148 pin no. 24 (P1.26/RTCK)(JTAG) *1
33	GND
34	LPC2148 pin no. 56 (P1.29/TCK)(JTAG) *1
35	GND
36	LPC2148 pin no. 52 (P1.30/TMS)(JTAG) *1
37	GND
38	LPC2148 pin no. 60 (P1.28/TDI) (JTAG) *1
39	GND
40	LPC2148 pin no.20 (P1.31/TRST)(JTAG) *1
41	3V3
42	3V3
43	Reserved for future use
44	TSOP 1738 Output pin via J3
45	Pins 45-60 Reserved for future use
46	
47	
48	
49	
50	
51	
52	

53	
54	
55	
56	
57	
58	
59	

60	
61	GND
62	
63	3V3
64	
65	Not connected
66	

Table 8.11: ARM7 LPC2148 microcontroller board expansion header table

Note:

*1. These pins can be used for JTAG interface as well as GPIO.
*2. These pins can be used as SPI bus for interfacing external devices.
*3. These pins can be used for interfacing external I2C devices

Figure 8.50 LPC2148 Expansion Socket Schematic

8.7.5 SERVO POD Socket

Fig 8.51: Servo Pod Socket

Figure 8.52 Servo pod pin mapping

Pin No.	Pin Name	Description
1	Servo POD1	Connection with LPC2148 ADC Sharp2
2		
3	Servo POD2	Connection with LPC2148 ADC Sharp4
4		
5	GPIO	Connection with LPC2148 P1.28/TDI
6		
7	Not Connected	Reserved for future use
8		
9	Not Connected	Reserved for future use
10		
11	Ground	Ground
12		
13	V SYS	+ 5V (VCC)
14		
15	V BATT	Battery Voltage(9V – 11V)
16		

Table 8.12: Servo Pod socket pin description

8.7.6 Microcontroller Board Jumpers

Microcontroller board has 3 jumpers labeled from J1 to J3 as shown in Figure 8.45

J1: Jumper J1 is used to enable/disable buzzer circuit on the main board. In the default state J1 is always inserted.

J2: Jumper J2 is used to connect the TSOP1738 output to pin no. 20 (P1.31) of the LPC2148 microcontroller only if jumper J3 is in its place as shown in Figure 8.48. This pin is used to sense the edge of incoming pulses from the TSOP1738 IR receiver. This pin is also used as TRST pin for JTAG interface in debug mode. If JTAG interface is used, remove jumper J2 to avoid interference from TSOP1738. In the default state J2 is kept open.

J3: Jumper J3 connects TSOP1738 output to pin no. 15 (P0.30) of the LPC2148 microcontroller as shown in Figure 8.48. This pin is used to measure the width of the incoming pulses from TSOP1738 IR receiver. In the default state J3 is kept open.

8.7.7 Boot/Interrupt Switch

The boot/interrupt switch is connected pin no. 41 of LPC2148 microcontroller as shown in Figure below. The purpose of this switch is to make LPC2148 enter in to boot loader mode if it is kept pressed on reset. Otherwise it can be used to interrupt LPC2148 microcontroller in application mode.

Figure 8.53 Interrupt switch

8.7.8 Servo Connectors

The microcontroller board has single Servo connector as shown in figure 8.45. It can be used for driving servo motor of camera pod or any other attachment. Power for the servo connector is provided by the “5V servo supply” regulator.

Figure: 8.54 Servo Connector schematic

8.7.9 Position Encoder LEDs

These LEDs are used to display the pulses coming from the left and right motor position encoders.

Figure 8.55: Encoder LEDs Schematic

8.7.10 WL 1, 2 & 3 Zener Diodes

The ADC module of LPC2148 microcontroller is operating at 3.3V. However the white line sensors are operating at 5V and its output voltage at certain point tends to go beyond 3.3volts. As this condition is undesirable by LPC2148 microcontroller a set of 3.3V zener diodes are connected to its ADC pins as shown in Figure 8.56 to clip any signal above 3.3 volts thereby compromising with the operating range of the sensors.

Figure 8.56 WL 1, 2 & 3 Zener Diodes

8.7.11 IR proximity 2 & 4

IR proximity sensors 2 & 4 and Sharp IR range finder 2 & 4 share same ADC pins on the LPC2148 microcontroller. Due to this only one type of the sensor can be used in the application. The Sharp sensors are connected directly to their ADC pins. IR proximity sensors are connected through 0E resistors as shown in Figure below. In default configuration the 0E resistors are always connected on the board so as to select IR proximity sensors. If you wish to use Sharp IR range finder 2 or 4 or both then remove their respective 0E resistors. Refer Figure 8.58 to locate the 0E resistors. The IR proximity sensors operate at 5V sensor supply whereas LPC2148 ADC module operates at 3.3 volts. To protect the ADC pins from being damaged, a resistor divider network is used to scale down analog signal from IR proximity to 3.3V.

Figure 8.57 IR Proximity2 & 4 Zener Diodes

Figure 8.58 0E Resistors

8.7.12 ARM7 LPC2148 Microcontroller

ARM7 LPC2148 microcontroller is interfaced directly to most of the onboard peripherals. Its schematic is shown in the figure 8.59.

Figure 8.59: LPC2148 microcontroller schematic

9. Upgrading Robot's Hardware

Figure 9.1: Fire Bird V robot and its accessories

Figure 9.2: Fire Bird V main board

9.1 Installing XBee wireless module

Figure 9.3: Mounting the XBee wireless module

⚠ Mount the ZigBee wireless module in the proper orientation.

⚠ You might have to configure ZigBee wireless module to the appropriate settings using USB wireless module and X-CTU software. For more details refer to Application notes.

9.2 Setting correct jumper settings on the main board

Figure 9.4: Set jumpers as per the requirements (for more details refer to chapter 3)

9.3 LCD mounting

Figure 9.5: LCD mounting

⚠ Be careful while inserting LCD connector pins into the socket on the main board. Screw in the LCD firmly on the studs.

5.4 Microcontroller adaptor board mounting

Figure 9.6: ATMEGA2560 (AVR), P89V51RD2 (8051) and LPC2148 ARM7 microcontroller adaptor boards for Fire Bird V

Figure 9.7: Microcontroller adapter board mounting

⚠ Do not apply unnecessary pressure onto the PCB while inserting into the connectors on the main board. Check for any bent pins before inserting the PCB. Mount 3 screws on the microcontroller board.

9.5 Sharp IR Range sensor mounting

Step 1: Cut the front section of white connector of the Sharp IR range sensor

Figure 9.8: Remove front cover of the white connector of the Sharp IR range sensor

Figure 9.9: Sharp IR range sensor mounting kit

Step 2: Mount the 20mm studs on the main board on the position where sharp sensor is to be fitted. In figure 5.10 area highlighted with the red border shows the mounted studs from the Sharp IR range sensor mounting kit.

Figure 9.10: Mount 20mm studs from the Sharp sensor mounting kit

Step 3: Fix the Sharp IR range sensor on its holder.
Holder for the Sharp IR range sensor is highlighted in the figure 5.10 by blue border.

Figure 9.11: Mounting of the Sharp IR range sensor

Step 4:
Remove the yellow colored paper stripe from the adhesive tape from the metal plate which is shown in figure 5. 9.
Fix the metal plate on top of the Sharp sensor and fit the screws.

Figure 9.12: Fitting metal plate on top of the Sharp IR range sensor

⚠ Do not apply extreme pressure while pressing down the sharp sensors to fit into the socket.

⚠ Make sure that you remove yellow paper before mounting the metal plate on the Sharp IR range sensor.

9.6 Mount top Acrylic plate on the robot

Figure 9.13: Install top acrylic plate