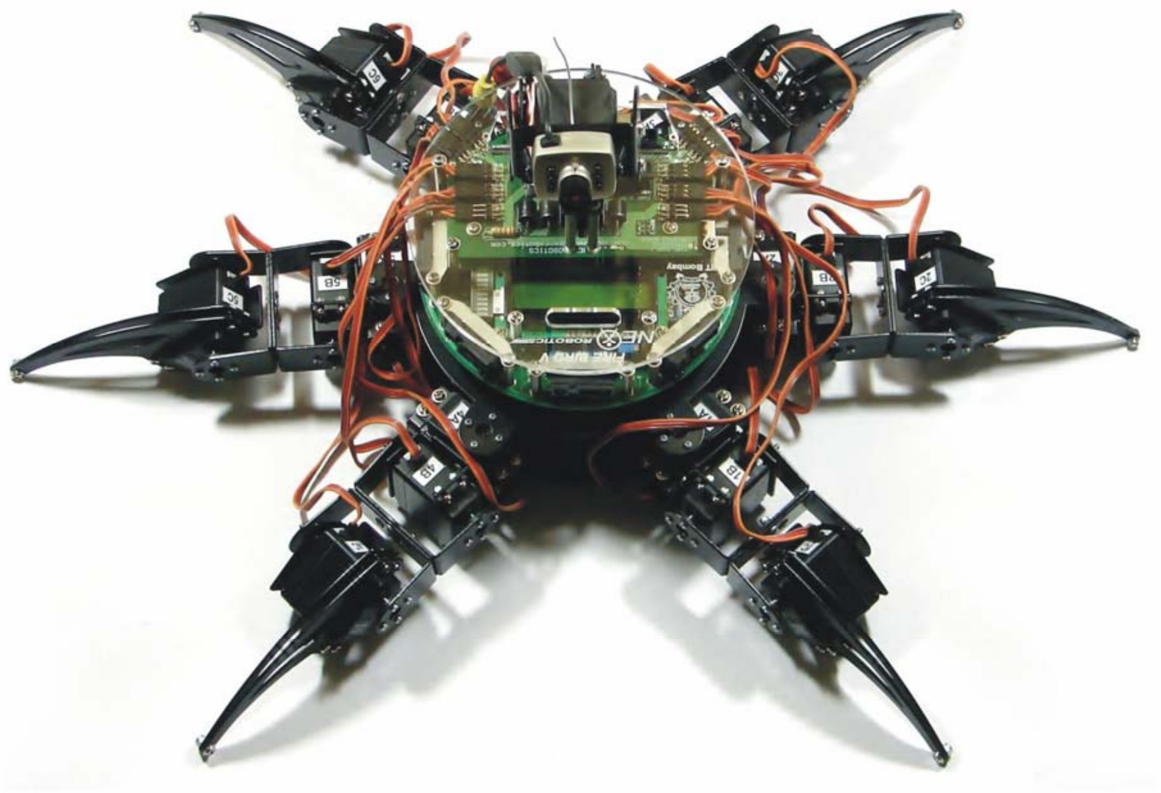
FIRE BIRD V

ATMEGA2560 HEXAPOD ROBOTIC RESEARCH PLATFORM USER GUIDE



Designed By:

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

NEX ROBOTICS
www.nex-robotics.com

Manufactured By: NEX Robotics Pvt. Ltd.

FIRE BIRD V

HEXAPOD ROBOT
USER GUIDE

Version 2.00
December 3, 2010

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

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

Notice

⚠Warning

Fire Bird V Hexapod Robot uses Lithium Polymer Battery. Refer to this user guide for the battery handling and charging instructions.

ATTENTION
OBSERVE PRECAUTIONS FOR HANDLING
ELECTROSTATIC SENSITIVE DEVICES

⚠ **Robot's electronics is static sensitive. Use robot in static free environment.**
⚠ **Read the hardware and software manual completely before start using this robot**

Recycling:

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

Important:

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

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

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

Index

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 Lithium Polymer battery only with the charger provided with the robot.
- Never allow Lithium Polymer battery to deep discharge. Charger will not charge deep discharged battery.
- 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.

⚠Warning

Fire Bird V Hexapod Robot uses Lithium Polymer Battery. Refer to this user guide for the battery handling and charging instructions.

If robot is used even when battery low indicator buzzer is on, it will cause the battery to deep discharge. In this case, the battery charger will not charge batteries for safety reasons

2. Fire Bird V ATMEGA2560 Hexapod Robot

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

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

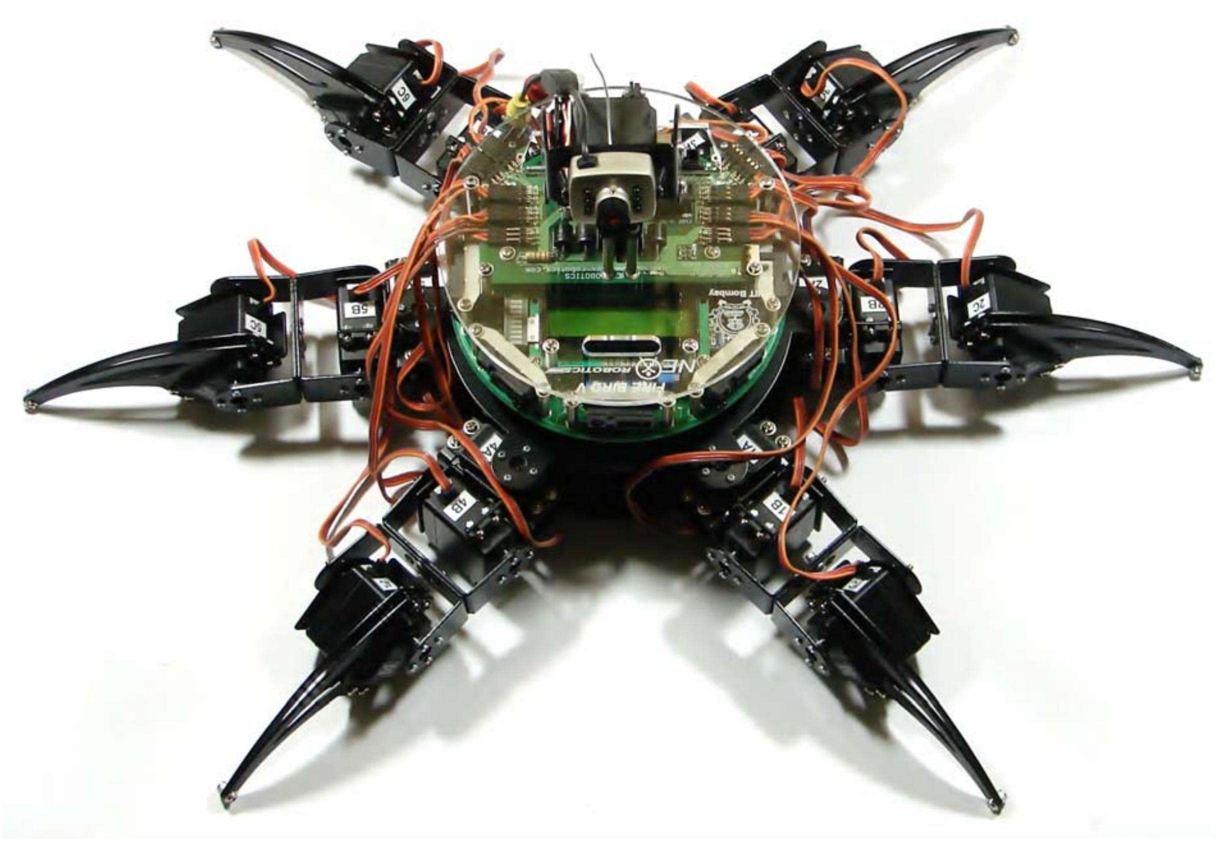


Figure 2.1: Fire Bird V Hexapod Robot

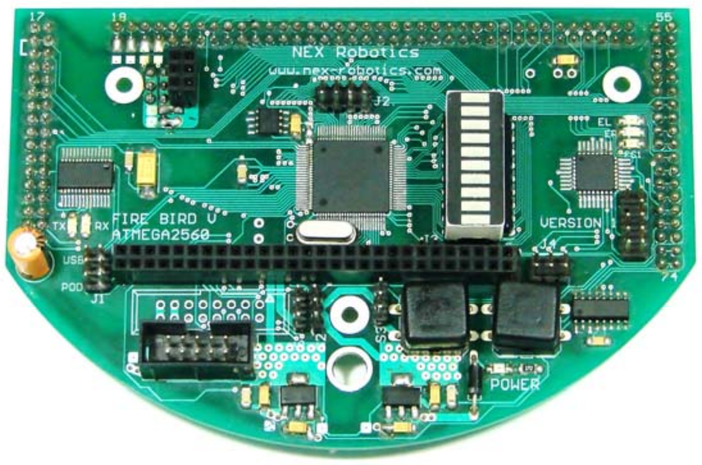


Figure 2.2: ATMEGA2560 (AVR) microcontroller adaptor board

2.1 Fire Bird V ATMEGA2560 technical specification:

Note:
All the features of the Fire Bird ATMEGA2560 Hexapod Robot except the Battery, Battery charging and the locomotion are exactly same as of Fire Bird V standard model. Use this user guide along with the robot's hardware and software manual.

Microcontroller:
Atmel ATMEGA2560 as Master microcontroller (AVR architecture based Microcontroller)
Atmel ATMEGA8 as Slave microcontroller (AVR architecture based Microcontroller)

Sensors:
Five Sharp GP2D12 IR range sensor (One in default configuration)
Eight analog IR proximity sensors
Eight analog directional light intensity sensors
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)
USB Communication
Wired RS232 (serial) communication
Simplex infrared communication (From infrared remote to robot)

Dimensions:

Height: 13cm (Base is on the ground)
Height: 19cm (Standing)
Diameter: 56cm (All the legs spread out)
Diameter: 44cm (Standing)
Weight: 2540gms

Power:

7.4V, 1800mAh Lithium Polymer battery (NR-BLiPo-2-1800) with smart battery charger NR-BLIC-03

Battery Life:

20 minutes while motors are operational at 75% of time

Locomotion:

Eighteen high torque, metal gear Servo motors (NRS-995) driving six legs (3 motors / leg)

3. Using Fire Bird V ATMEGA2560 Hexapod Robot

3.1 Powering up the robot

Fire Bird V ATMEGA2560 Hexapod has onboard rechargeable 7.4V, 1800mAh Lithium Polymer battery (NR-BLiPo-2-1800), it can power robot for approximately 20 minutes.

The Robot has power board for managing the power requirement for the robot. The power switch shown in figure 3.1 on the main board controls, power to the Firebird V's main board and ATMEGA2560 microcontroller adaptor board. The power to the main board is connected to battery connector of the FBV Main Board with the 10 pin FRC cable. Power to the all servos is taken directly from the power board to the servo control card via relay. This relay controls power to the servo motors. Figure 3.2 shows the power wires for the servo control board.

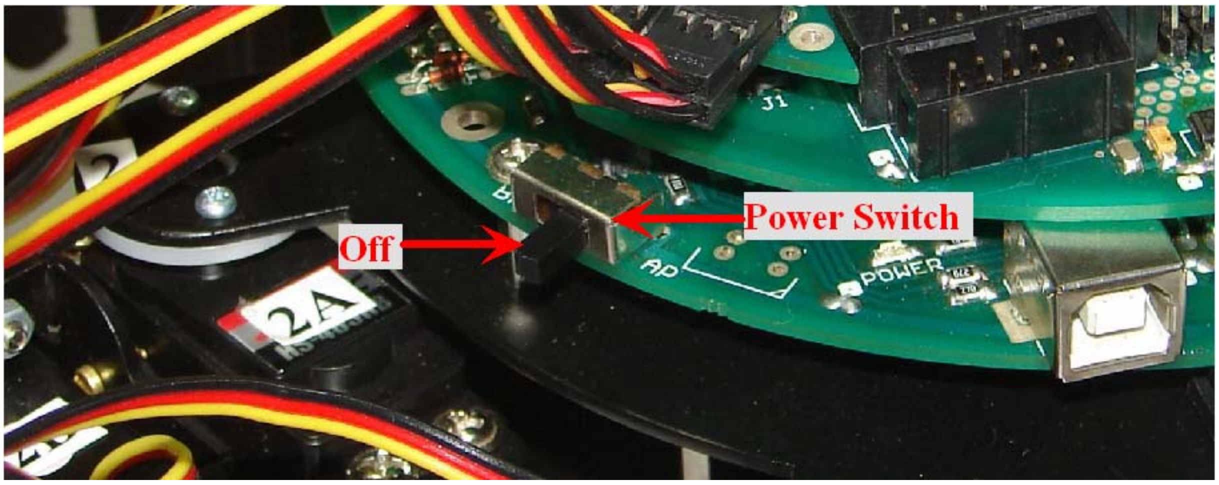


Figure 3.1 Power Switch on FB V Main Board

For the safety during the transportation battery connection of the robot is disconnected. Before connecting battery with the robot make sure that power switch is in off state. Hold the robot in your hand and then turn it on. After robot's legs are initialized in standing position, gently put it down on the ground. To urn off the robot, hold it in your hands, turn the power off and then put it on the ground. Make sure that robot's base is touching the ground when its power is off.

Important:

- If you are using robot after a week or so, charge robot's battery before using. Battery charging is covered in the section 3.2.
- If you are using the robot for the first time, please go through the manual carefully before turning on the robot.
- To turn the robot on or off follow the exact sequence as mentioned in the section 3.1. Follow this sequence to extend the robot's life.

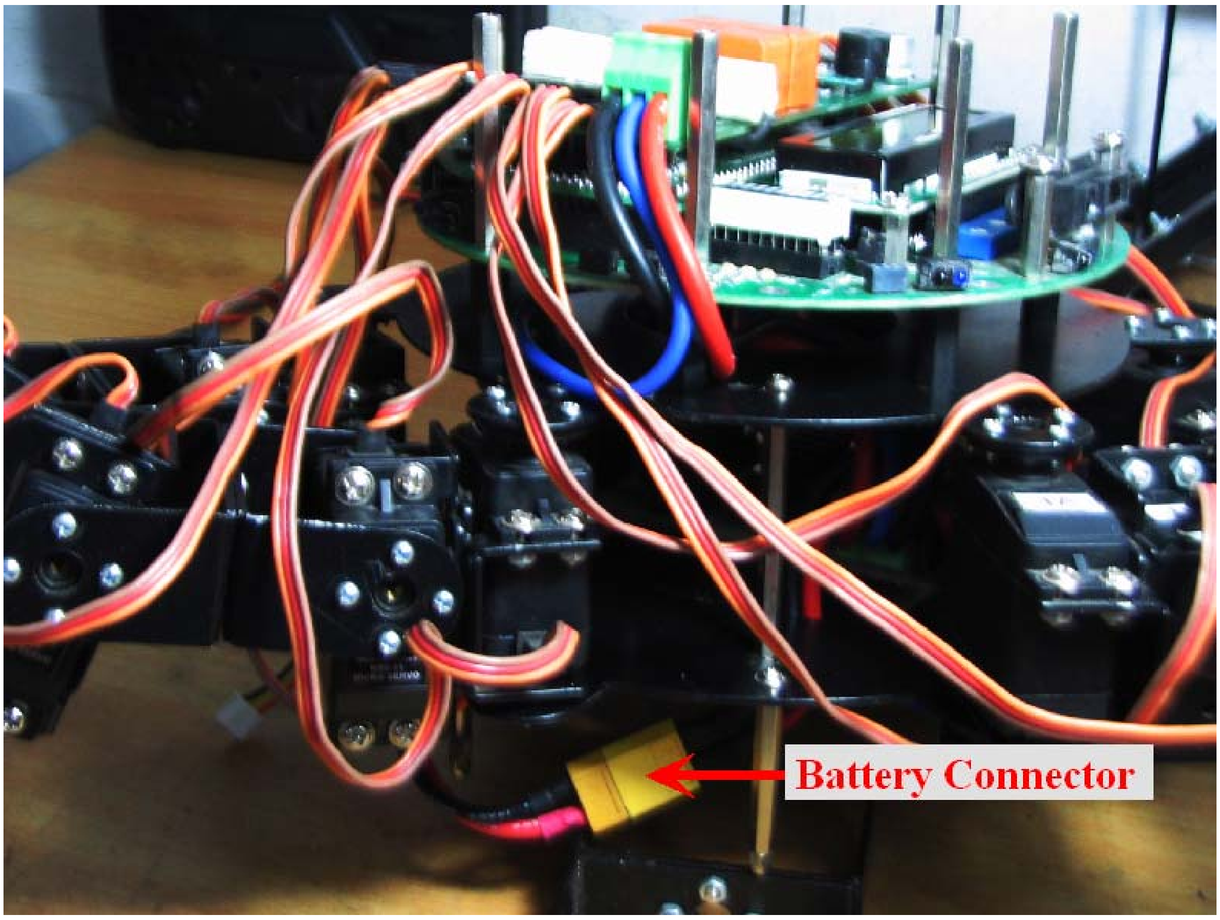


Figure 3.2: supply connection on the Hexapod

3.2 Battery Charging

Fire Bird V ATMEGA2560 Hexapod robot requires higher current than standard configuration robot. In order to fulfill this additional current requirement while keeping weight low, 7.4V, 1800mAh lithium polymer battery (Product code: NR-BLiPo-2-1800) is used. It can supply discharge current up to 36 Amps.

Battery is fixed inside the battery holder by means of velcro strap. To charge the battery remove it from the Battery holder and charge it externally. It is highly recommended that you charge the batteries in the open environment outside the building.

Never allow battery to go below 6.6V. Robot’s battery low indication threshold is set at 6.6V. When battery voltage drops below 6.6V, Battery buzzer will start beeping.

In multi-celled Lithium Polymer battery pack, it is possible for the individual cells to develop differences in there charge levels. Since Lithium Polymer batteries are very sensitive to overcharging, it is important that cells inside the battery pack should be kept at the equal levels when charging.

Fire Bird V ATMEGA2560 Hexapod robot comes with its own NR-BLIC-03 balance charger form the NEX Robotics. To charge the battery, remove it from the robot and charge it in the open space. Figure 3.4. shows the battery charging process. NR-BLIC-03 balance charger monitors individual cell voltages of the battery pack and it adjusts the rate of charge to the individual battery to do balance charging. Use only NR-BLIC-03 battery charger for battery charging.

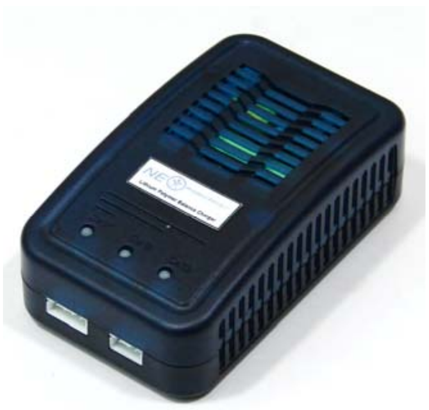


Figure 3.3: NR-BLIC-03 battery charger from NEX Robotics

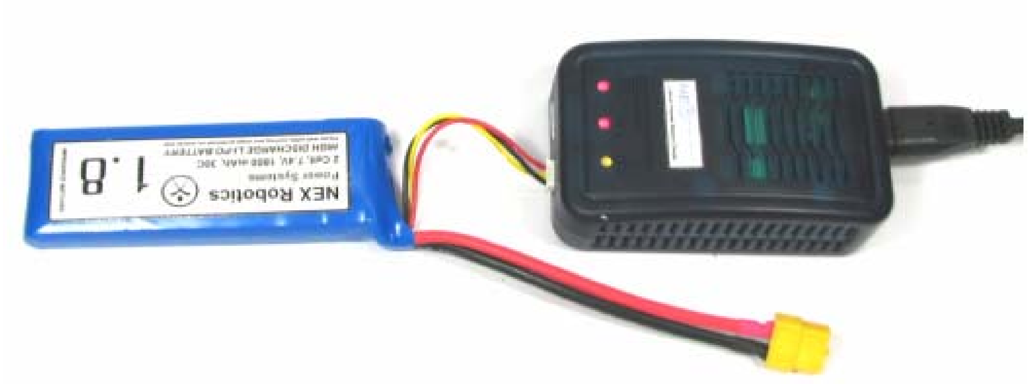


Figure 3.4 Connecting battery with NR-BLIC-03 charger

⚠Warning

Never ever charge the battery while the robot is ON. It will damage battery or charger or both.

Instructions for the battery charging:

For battery charging instructions refer to “Lithium Polymer Balance Charger NR-BLIC-03, NEX Robotics.pdf” which is located in the “Manuals and Application notes” folder in the documentation CD.

Two cell and three cell battery packs from the NEX Robotics has different types of connectors. These batteries will go only in the correct type of connector of the Battery Charger.

Warning:

- Do not charge 2 cell and 3 cell batteries at the same time.
- Charge batteries which can handle 650mA charging current.
- If battery is hot or slightly warm then allow it to cool down completely before charging.
- Do not open battery packs and modify them. Modified packs will not have matched impendence, which can lead to dangerous situations.
- Always charge batteries in open space of at least 10feet x 10feet on the concrete floor.
- Do not charge battery near flammable liquids.
- While charging put batteries away from children.

Important:
Never ever allow battery to discharge below 3.3V per cell i.e. 6.6V for 2 cell battery packs or 9.9V for 3cell battery packs. After this critical value battery voltage falls very rapidly. Robot will start giving battery low warning after battery voltage reaches 6.6V. Battery charger will not charge any deep discharged battery pack and it will indicate fault condition. This is done for the safety reasons.

3.3 Running the robot

Servo motors are quite delicate. To protect them from overload take robot in your hand then turn it on. Once legs are powered put it down. When program is being loaded on the robot, its legs can make sudden moves. . If you are not sure about your code, put the robot on the CD box before turning it ON or while programming, to avoid damage to the servo motors.

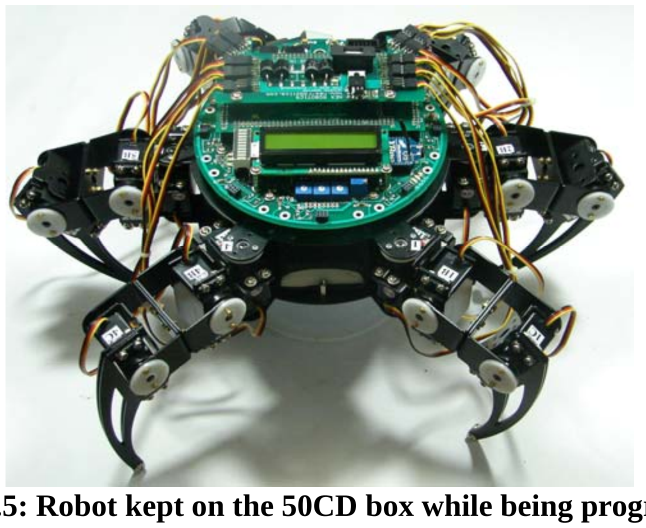


Figure 3.5: Robot kept on the 50CD box while being programmed

3.4 Using Remote Control

Fire Bird V Hexapod can be controlled using Modified RC Remote control provided with the robot. Remote control has two analog joysticks having 2 axis each and two user programmable switches. Remote control has XBee wireless module and ATMEGA8 microcontrollers. Its source code comes with its documentation. Remote control's firmware can be upgraded via its 10 pin ISP cable. It requires AA size 8 batteries. You can use ordinary pencil cells or rechargeable NiMH batteries. If battery voltage falls below critical value it will start giving beeping sound.

To control the robot using remote control, first load "Hexapod_remote_control.hex" on the robot. It is located in the "Experiments / Fire Bird V ATMEGA2560 Hexapod folder in the documentation CD.

During the remote control operation robot acts as master while remote acts as slave. Robot asks for the data related to joystick's position, switch's logic level, battery voltage etc. and remote transmits the data. When robot and remote both are powered up once proper communication link is established robot and remote both beeps twice.

You can use right joystick to move robot forward, back, clockwise and counterclockwise direction. Left joystick is used to move servo pod mounted camera pod in pan and tilt direction. If any direction is given to the robot then robot will complete one set of motion and then only respond to the new motion direction. Robot needs maximum of 2 seconds to complete one motion sequence.

For more details on modified RC remote control refer the "RC Remote Control.pdf" which is located in the "Manuals and Application notes\Manuals\Fire Bird V ATMEGA2560 Hexapod Robot" folder in the documentation CD.

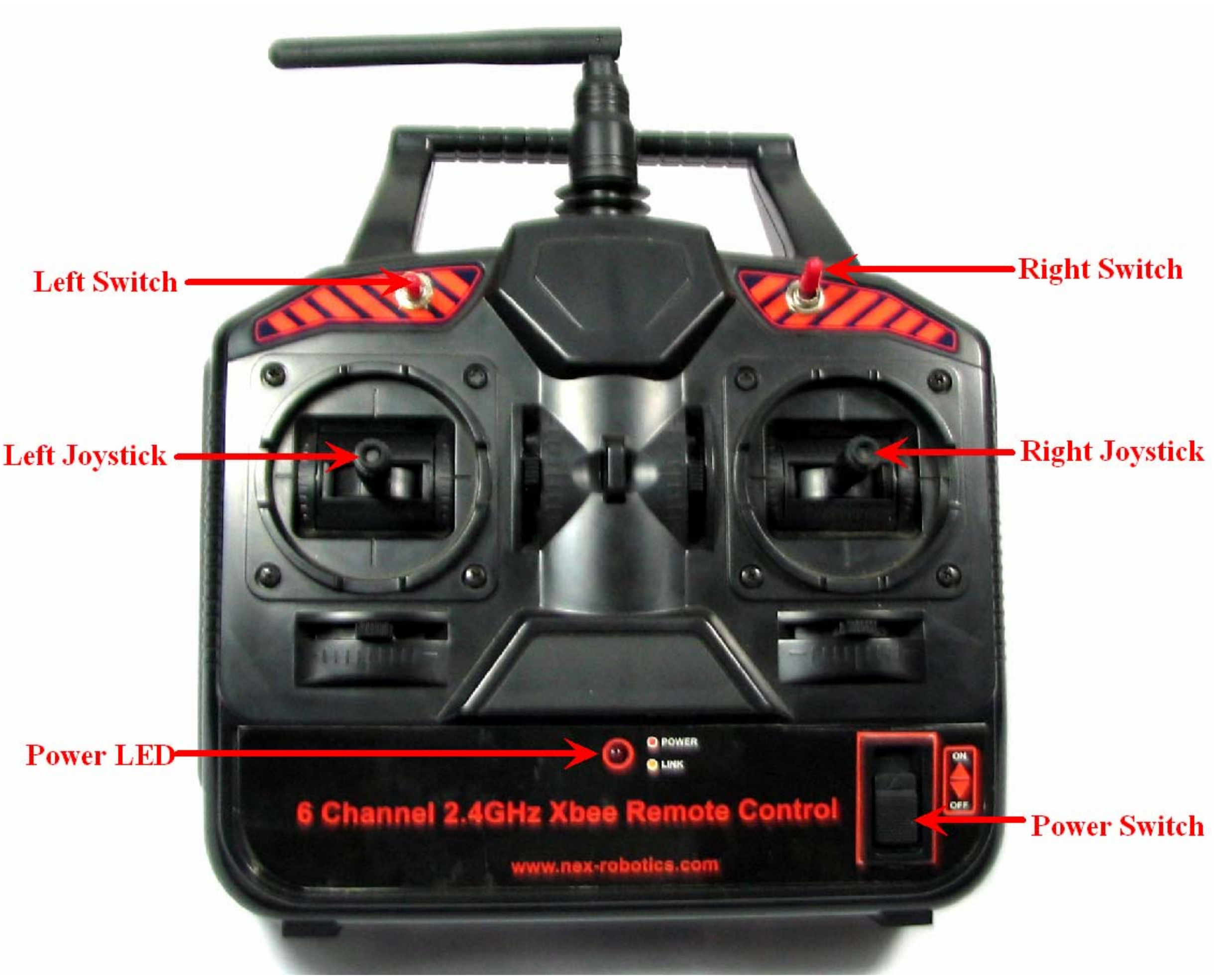


Figure 3.6: Modified RC remote control for the robot

3.5 Loading firmware on the robot

Programming of the robot is done through In-System Programming (ISP) using external ISP programmer or via bootloading. For more details on loading the firmware on the robot, refer to chapter 2 of the Software Manual.

There are two examples for the hexapod robot. They are located in the “Experiments / Fire Bird V ATMEGA2560 Hexapod Robot” folder in the documentation CD.

1. Hexapod_locomotion
This demo code runs robot in Forward, Back, Left and Right direction.

2. Hexapod_remote_control
Use this demo code to control robot and its camera pod (if installed).

4. Hardware Description

This chapter contains the hardware description of the Fire Bird V ATMEGA2560 hexapod robot.

Fire Bird V hexapod robot platform has 4 boards:

1. Power board
2. Main board for the Fire Bird V
3. ATMEGA2560 microcontroller adaptor board for the Fire Bird V
4. Servo control board

4.1 Power board

Power board is mounted inside the robot chassis. It has built-in fuse and fuse blown LED indicator at the front side. If fuse is blown, this red LED will glow. Power board gives 7.4V to the Robot's main board and regulated 6V, 20Ampp to the servo motors.

Note:
If Fuse blown LED glows then replace the fuse and try to run the robot again. If fuse is blow again then look for probable cause of short circuit. Use any standard 20A fuse as replacement.

4.2 Main board for Fire Bird V

Main board is a standard main board used in all the Fire Bird V series of robots. For more details on the hardware, refer to Hardware Manual.

4.3 ATMEGA2560 microcontroller adaptor board for the Fire Bird V

ATMEGA 2560 Microcontroller is mounted on the main board. It's a standard board used in all Fire Bird V robots. For more details on the hardware, refer to Hardware Manual.

4.4 Servo control board

Servo control board is mounted on the ATMEGA2560 microcontroller board. This board is used to connect / interface all 18 servo motors of the hexapod to the ATMEGA2560 microcontroller board. It has relay used as switch to control power given to servo motors. Relay is connected to the PORTD of the ATMEGA2560 microcontroller. It also has battery voltage monitoring circuit. Figure 4.1 shows the connections of the servo control board.

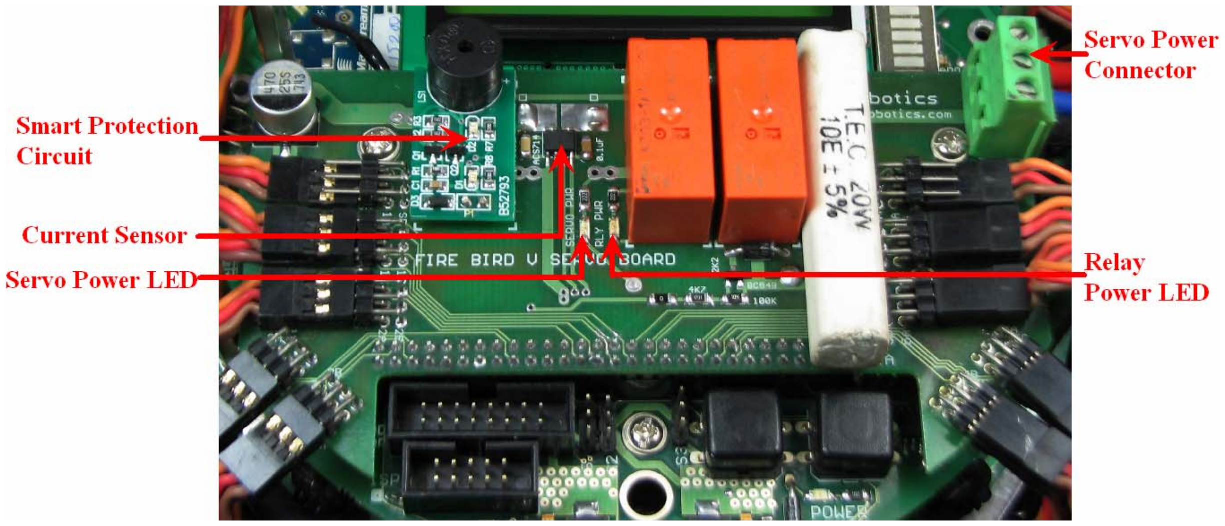


Figure 4.1 Servo Control Board

4.4.1 Servo connection and supply

Each servo motor requires Vcc, Ground and position data encoded in the form of PWM. Servo comtrol card can interface 20 servo motors at a time. Power given to these servo motors can be turned on/off by controlling relay which is connected to the PD0 of the ATMEGA2560 microcontroller. When relay is turned on, “RLY PWR” LED is also turns on. “SEVRO PWR” LED indicates power across servo motors.

4.4.2 Smart Battery Protection Circuit

Smart Protection Circuit board shows battery level in 4 steps and gives audio warning if battery is discharged below critical level.

Indications on Smart Protection Circuit Board:
Battery voltage above 7.41V: Blue LED on
Battery voltage between 7.4V and 6.8V: Blue LED blinks
Battery voltage between 6.8 and 6.6V: Red LED blinks
Battery voltage below 6.6V: Red LED blinks and Buzzer beeps

4.4.3 Current sensor

ACS714, 20Amp Hall Effect current sensor is used to measure current consumed by the 18 servo motors. It is an optional accessory. It gives out 100mV per ampere of current. Analog output of the current sensor is connected to the PORTF 5 (ADC5) pin of the microcontroller. ADC5 is also connected to IR Proximity sensor 2. In order to sense the current sensor’s output you need to disconnect IR Proximity sensor 2 from the ADC5. This can be done by removing 2nd jumper from the left of the J2 on the microcontroller adaptor board. For location of the J2 refer to figure 3.58 in the Hardware Manual.

4.5 Hexapod legs naming convention

Fire Bird V Hexapod has six legs. Each leg has three motors. To avoid confusion while addressing motors certain naming convention is used. Starting from the top left each leg is named from 1 to 6. Three motors on the each legs are named as A, B or C, i.e. each motor have the unique name as 1A or 3C etc.

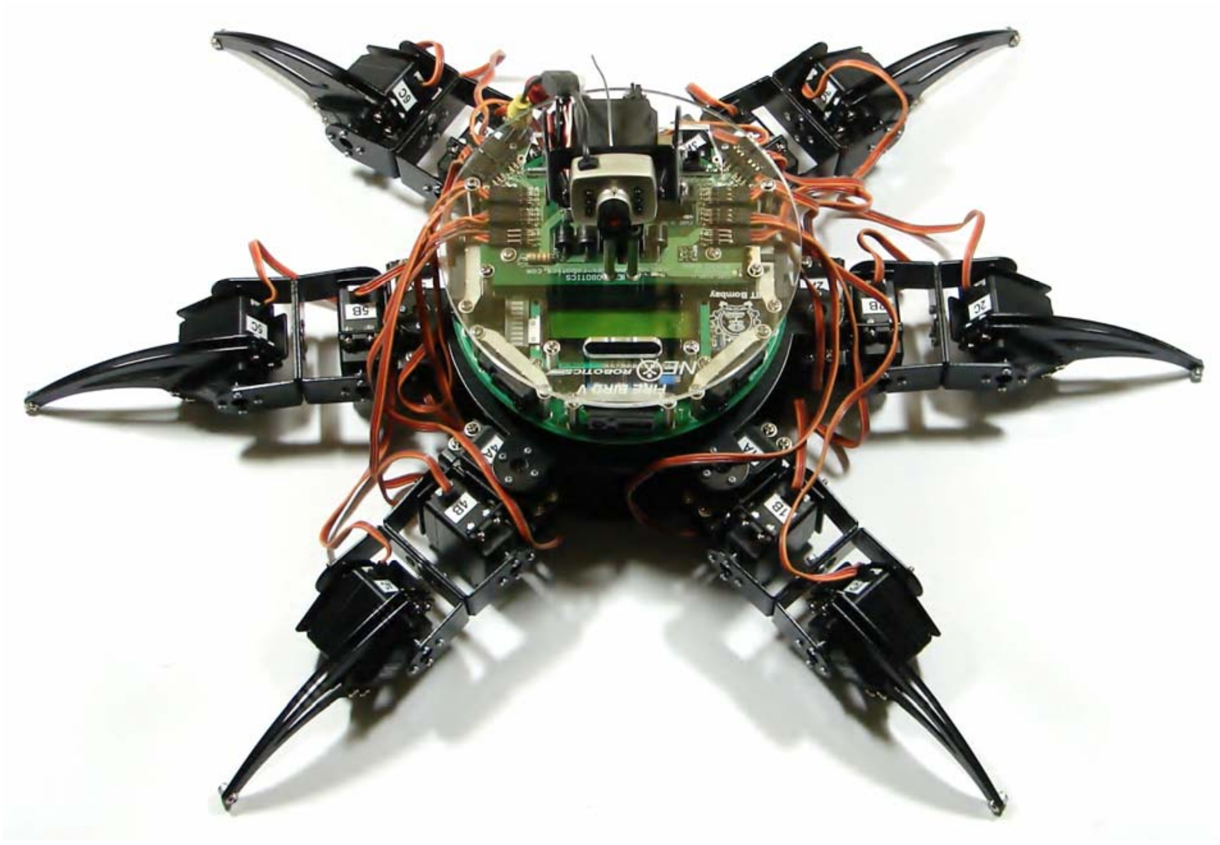


Figure 4.2: Fire Bird V Hexapod motor naming convention (stickers on the each servo motor)

4.5 Servo Connectors for camera mounted servo pod

ATMEGA 2560 microcontroller board has three Servo connectors. It can be used for driving servo motors of camera pod for tilt and pan motion. Connect tilt servo to the S1 socket and pan servo to the S2 socket. Servo connections are shown in the figure 4.3.

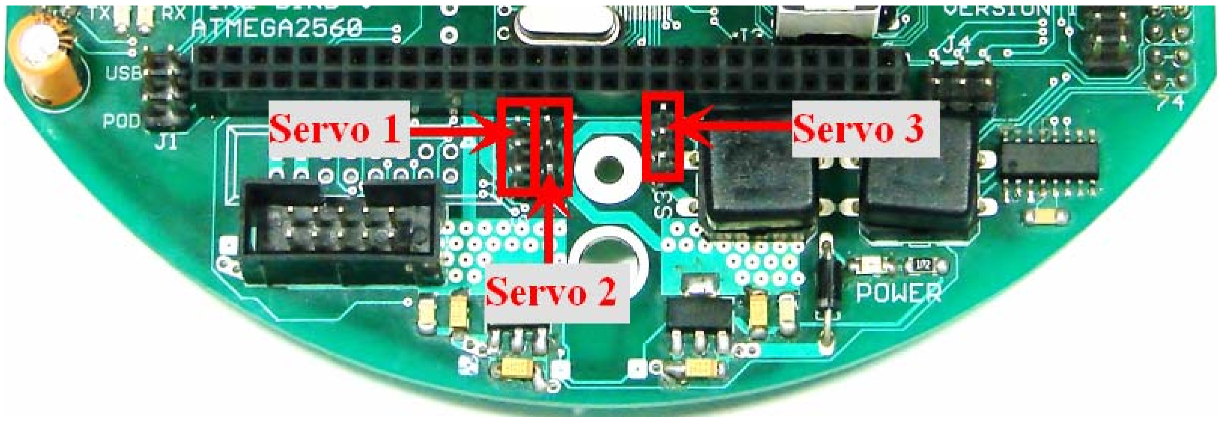


Figure: 4.3 Servo Connectors

4.6 ZigBee wireless communication for remote control application

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

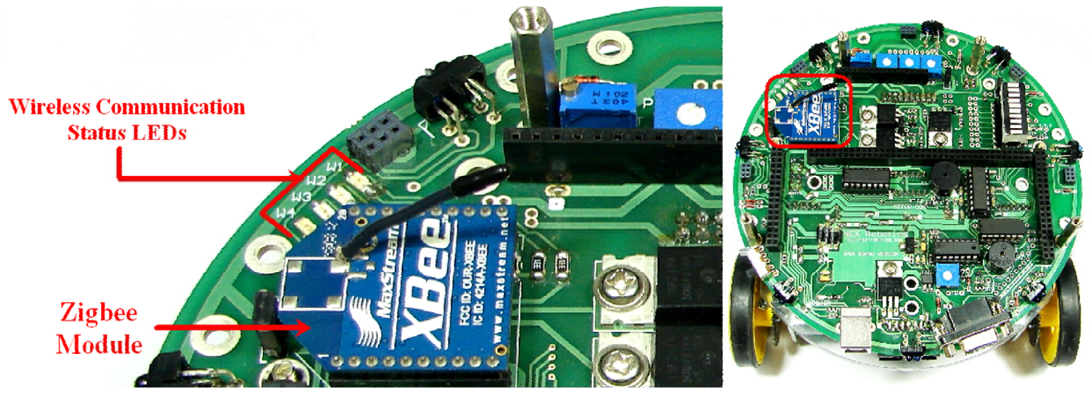


Figure 4.4: ZigBee wireless module and LED indicators

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 4.1: ZigBee LED Description

5. Microcontroller Pin Functionality

Following table shows the pin functionality of the Fire Bird V.

Pin No.	Pin name	USED FOR	Status
1	(OC0B)PG5	Slave Select (SS) of the SPI expansion port on the main board (refer to figure 8.5)	(unused)
2	RXD0/PCINT8/PE0	UART 0 receive For ZigBee wireless module (if installed)	Default
3	TXD0/PE1	UART 0 transmit For ZigBee wireless module (if installed)	Default
4	XCK0/AIN0/PE2	GPIO* (Available on expansion slot of the microcontroller socket)	unused
5	OC3A/AIN1/PE3	---	Output(unused)
6	OC3B/INT4/PE4	---	---
7	OC3C/INT5/PE5	---	---
8	T3/INT6/PE6	--	---
9	CLK0/ICP3/INT7/ PE7	---	Input(unused)
10	VCC	---	--
11	GND	---	--
12	RXD2/PH0	---	Default
13	TXD2/PH1	UART 2 transmit for USB Communication or Boot loading	Default
14	XCK2/PH2	Sharp IR ranges sensor 1and 5 disable. Turns off these sensors when output is logic 1 *******	Output(unused)
15	OC4A / PH3	IR proximity sensors 1 to 8 disable. Turns off these sensors when output is logic 1 *******	Output(unused)
16	OC4B / PH4	Servo 3B	Output
17	OC4C / PH5	Servo 3C	Output
18	OC2B / PH6	Servo 3A	Output
19	SS/PCINT0/PB0	ISP (In System Programming), SPI Communication with ATMEGA8 **, Connection to the SPI port on the main board	Output
20	SCK/PCINT1/PB1		Output
21	MOSI/PCINT2/PB2		Output
22	MISO/PCINT3/PB3		Input
23	OC2A/PCINT4/PB4	Servo Pod GPIO	(unused)
24	OC1A/PCINT5/PB5	PWM for Servo motor 1. *** (tilt servo camera pod)	Output
25	OC1B/PCINT6/PB6	PWM for Servo motor 2. *** (pan Servo camera pod)	Output
26	OC0A/OC1C/PCINT7/ PB7	PWM for Servo motor 3. ***	Output(unused)
27	T4/PH7	GPIO (Available On Expansion Slot)	(unused)
28	TOSC2/PG3	RTC (Real Time Clock)****	(unused)
29	TOSC1/PG4		(unused)
30	RESET	Microcontroller reset	--
31	VCC	5V	--
32	GND	Ground	--
33	XTAL2	Crystal 14.7456 MHz	--
34	XTAL1		
35	ICP4/PL0	GPIO (Available on expansion slot of the microcontroller socket)	(unused)
36	ICP5/PL1	Left side spare servo	--
37	TS/PL2	Servo 1B	Output
38	OC5A/PL3	---	---
39	OC5B/PL4	---	---
40	OC5C/PL5	---	---

41	PL6	Servo 2A	Output
42	PL7	Servo 2C	Output
43	SCL/INT0/PD0	Relay ON/OFF	Output
44	SDA/INT1/PD1	Available on expansion slot of the microcontroller socket	unused
45	RXD1/INT2/PD2	UART1 receive for RS232 serial communication	Default
46	TXD1/INT3/PD3	UART1 transmit for RS232 serial communication	Default
47	ICP1/PD4	Servo 2B	Output
48	XCK1/PD5	Servo 1A	Output
49	T1/PD6	Servo 4C	Output
50	T0/PD7	Servo 4B	Output
51	PG0/WR	Servo 4A	Output
52	PG1/RD	Servo 1C	Output
53	PC0	LCD control line RS (Register Select)	Output
54	PC1	LCD control line RW(Read/Write Select)	Output
55	PC2	LCD control line EN(Enable Signal)	Output
56	PC3	Buzzer	Output
57	PC4	LCD data lines (4-bit mode)	Output
58	PC5		
59	PC6		
60	PC7		
61	VCC	5V	
62	GND	Ground	
63	PJ0/RXD3/PCINT9	LED bargraph display and GPIO*	Output
64	PJ1/TXD3/PCINT10	Servo 6B	
65	PJ2/XCK3/PCINT11	Servo 6C	
66	PJ3/PCINT12	Servo 6A	
67	PJ4/PCINT13	Servo 5C	
68	PJ5/PCINT14	Servo 5A	
69	PJ6/PCINT15	Servo 5B	
70	PG2/ALE	Sharp IR ranges sensor 2, 3, 4 and red LEDs of white line sensor 1, 2, 3 disable. ******* Turns off these sensors when output is logic 1	Output(unused)
71	PA7 C2-2	---	---
72	PA6 C2-1	---	---
73	PA5 C1-2	---	---
74	PA4 C1-1	--	--
75	PA3	---	---
76	PA2	---	---
77	PA1	---	---
78	PA0	---	---
79	PJ7	Right side spare servo	--
80	VCC	5V	--
81	GND	Ground	--
82	PK7/ADC15/PCINT23	ADC Input For Servo Pod 2	Input (unused)
83	PK6/ADC14/PCINT22	ADC Input For Servo Pod 1	Input (unused)

84	PK5/ADC13/PCINT21	ADC input for Sharp IR range sensor 5	Input (unused)
85	PK4/ADC12/PCINT20	ADC input for Sharp IR range sensor 4	Input (unused)
86	PK3/ADC11/PCINT19	ADC input for Sharp IR range sensor 3	Input (unused)
87	PK2/ADC10/PCINT18	ADC input for Sharp IR range sensor 2	Input (unused)
88	PK1/ADC9/PCINT17	ADC input for Sharp IR range sensor 1	Input (unused)
89	PK0/ADC8/PCINT16	ADC input for IR proximity analog sensor 5	Input (unused)
90	PF7(ADC7/TDI)	ADC input for IR proximity analog sensor 4*****	Input (unused)
91	PF6/(ADC6/TD0)	ADC input for IR proximity analog sensor 3*****	Input (unused)
92	PF5(ADC5/TMS)	ADC input for current sense voltage *****	Input (Floating)
93	PF4/ADC4/TCK	ADC input for IR proximity analog sensor 1*****	Input (unused)
94	PF3/ADC3	---	Input (unused)
95	PF2/ADC2	---	Input (unused)
96	PF1/ADC1	---	Input (unused)
97	PF0/ADC0	ADC input for battery voltage monitoring	Input (unused)
98	AREF	ADC reference voltage pin (5V external) ******	Input (unused)
99	GND	Ground	--
100	AVCC	5V	--

Table 5.1 ATMEGA2560 pin functionality in Fire Bird V Hexapod

* Not used pins are by-default initialized to input and kept floating. These pins are available on the expansion slot of the ATMEGA2560 microcontroller adaptor board.

** MOSI, MISO, SCK and SS pins of ATMEGA2560 are associated to the ISP (In System programming) port as well as the SPI interface to ATMEGA8. J4 needs to be disconnected before doing ISP. To communicate with ATMEGA8 jumper J4 needs to be in place. Refer to section 2.6 for more details from ATMEGA2560 software manual.

*** PORTB pin5, 6, 7 are OC1A, OC1B, OC1C of the Timer1, that are connected to the servo motor sockets S1, S2, S3 on the microcontroller adaptor board.

**** External Crystal of 32 KHz is connected to the pins PG3 and PG4 to generate clock for RTC (Real Time Clock).

***** For using Analog IR proximity (1, 2, 3 and 4) sensors short the jumper J2. To use JTAG via expansion slot of the microcontroller socket remove these jumpers.

****** AREF can be obtained from the 5V microcontroller or 5V analog reference generator IC REF5050 (optional). Refer section 3.19.9 for more details from Fire Bird V TAMEGA2560 Hardware manual.

******* Sensor’s switching can be controlled only is if corresponding jumpers are open. Refer to section 3.11 and 3.12 from the hardware manual for more details.
J2: Sharp IR range sensor 2, 3, 4 and red LEDs of white line sensors;
J3: Sharp IR range sensor 1, 5;
J4: IR proximity sensors 1 to 8;

******** External interrupt from the position encoder C1 is disabled by removing pin 2 of the CD40106 Schmitt trigger inverter buffer to avoid its wire ANDing with the interrupt switch.

ATMEGA8 pin configuration

PIN NO	Pin name	USED FOR
1	INT1/PD3	Not Used
2	XCK/TOSC1/PB6	Not Used
3	GND	Ground
4	VCC	5V
5	GND	Ground
6	VCC	5V
7	XTAL1/TOSC1/PB6	Not Used
8	XTAL2/TOSC1/PB7	Not Used
9	(T1) PD5	Not Used
10	(AIN0) PD6	Not Used
11	(AIN1) PD7	Not Used
12	(ICP) PB0	Not Used
13	(OC1A) PB1	Not Used
14	(SS/OC1B) PB2	ISP (In System Programming) and SPI Communication with ATMEGA2560. *
15	(MOSI/OC2) PB3	
16	(MISO) PB4	
17	PB5 (SCK)	
18	AVCC	5V
19	ADC6	ADC input for IR proximity analog sensor 7
20	AREF	5V
21	GND	Ground
22	ADC7	ADC input for IR proximity analog sensor 8
23	PC0 (ADC0)	ADC input for white line sensor 4
24	PC1 (ADC1)	ADC input for white line sensor 5/Servo pod
25	PC2 (ADC2)	ADC input for white line sensor 6
26	PC3 (ADC3)	ADC input for white line sensor 7/Servo pod
27	PC4 (ADC4/SDA)	ADC input for Current Sensing IC ACS712
28	PC5 (ADC5/SCL)	ADC input for IR proximity analog sensor 6
29	PC6 (RESET)	Microcontroller reset
30	PD0 (RXD)	Not Used
31	PD1 (TXD)	Not Used
32	PD2 (INT0)	Not Used

Table 5.2: ATMEGA8 microcontroller pin connections

***** MOSI, MISO, SCK and SS pins of ATMEGA2560 are associated to the ISP (In System programming) port as well as the SPI interface to ATMEGA8. J4 needs to be disconnected before doing ISP. To communicate with ATMEGA8 jumper J4 needs to be in place. Refer to section 4.6 for more details in the ATMEGA2560 Hardware manual.

6. Application Example Code.

All codes for the robot and remote control are written in AVR Studio version 4.17. In this chapter various functions used in the hexapod are covered.

Important:

1. **While programming the robot keep it on the CD box.**
2. **Always call "robot_stand_position ()" function first, wait for at least 3 seconds and then start your motion.**
3. **Test your motion on the CD box first before starting execution.**

6.1 Servo control

Fire Bird V Hexapod uses High torque NRS-995 metal gear servo motors. This servo motor gives 0 to 180 degree rotation for the 0.5ms to 2.2ms pulse width. This control pulse train can be given to the motor at 40 to 60 Hz.

6.2 Servo port initialization

Required pins of the microcontroller for servo control application are configured in this function. For pin connection refer to table 5.1.

```
void servo_pin_config (void)
{
 DDRB = 0x60;     //PB 5,6 as output
 PORTB = 0x00;
 DDRD = 0xF1;    //PD 1,4,5,6,7 as output
 PORTD = 0x00;
 DDRG = 0x03;    //PG 0,1 as output
 PORTG = 0x00;
 DDRH = 0x70;    //PH 4,5,6 as output
 PORTH = 0x00;
 DDRJ = 0xFE;    //PJ 1,2,3,4,5,6,7 as output
 PORTJ = 0x00;
 DDRL = 0xC7;    //PL 1,2,3,6,7 as output
 PORTL = 0x00;
}
```

6.3 Servo control signal generation

We need to control 18 servo motors simultaneously. Each servo motor needs control signal at 40 to 50Hz repetition rate with 0.5ms to 2.2ms pulse width depending upon the desired position. In the conventional way of generating such signal, each timer can only generate three such outputs. Each output will only have resolution of about 1.5 degrees (10 bit). We do not have sufficient number of timers to control 18 servo motors. Hence Timer1 is used in a bit different way which enables us to control up to 24 servos at the same time. Only disadvantage is that this algorithm generates about 1600 interrupts per second. In the Fire Bird V clock is running at 14.7456MHz. ATMEGA2560 process almost all the instruction in single clock cycle. This means microcontroller gets interrupts after every 7000 instructions. Hence floating point calculation should be avoided at all cost as microcontroller is already quite loaded. If we are using this algorithm then use of LCD is not recommended. It might interfere with the stability of the code.

Algorithm:

1. Timer 1 gets overflowed at every 2.5ms (400Hz). It also generates Timer 1 overflow interrupt each time it gets overflowed **"ISR (TIMER1_OVF_vect)"**.
2. In the timer 1 overflow interrupt unsigned char variable "arm_number" (0 to 6) gets incremented every time ISR occurs. It can only incremented from 0 to 7 and then back to 0.
3. Function like set_1A () and reset_1A () are used to set or reset particular control pin of the servo motor.
4. When there is a particular arm number is in the ISR, pins corresponding to that arm are set to logic 1 using functions like **set_1A ()** where arm number can be from 1 to 6 and motor location from A to C. It means that each pin gets set only for 400Hz / 8 = 50 times a second. In this way control frequency of 50 Hz is generated for each servo motor.
5. Corresponding pins are reset to logic 0 in the OCR interrupts when the particular arm number is set by the timer 1 overflow ISR. In this way desired pulse width between 0.5ms to 2.2ms is obtained.

6.4 Servo angle control

Desired servo angle is calculated in the function **"angle_value_calculation ()"**.
"angle_value_calculation ()" calculates the 16 bit OCR value for the particular servo motor and separates upper and lower bytes which are to be loaded in to the corresponding OCRs for setting the corresponding servo angle. This function is used by all the functions such as **"angle_1A(unsigned char angle)"**to set individual angle. In this function constant of 139 is used instead of 139.4 which adds some error in the angle calculation but by using unsigned char instead of floating point number, lots of computational requirement is saved.

6.5 Servo motor calibration

This function is used for calibrating servo motors. If you need to replace any servo then call **"robot_arm_calibration ()"** function. All the motors are set in the 90 degrees standup position. Now any of the servos can be replaced. This function must be the first function to be called after **"init_devices ()"** in the main function to initialize the motors. If its is not called first then all the servos will assume 0 degree angle and all the arms will get twisted and robot might take sudden fall and servos might get damaged because of the impact. This function is also used frequently in various motion functions.

6.6 Robot forward and backward motion

"forward ()" and **"back ()"** functions moves robot forward and back. In the forward and back motion in each step a pair of legs from the both side is lifted and moved in the desired direction, call the function **"fwd_front_step ()"** for forward step or call **"back_front_step ()"** for back step. Once all the legs are moved, robot pushes itself in the desired direction by calling the function **"walk_stroke()"**. Same motion can be achieved as shown in the video "Hexapod 1" in the video folder which is located in the documentation CD.

6.7 Robot clockwise (CW) and counter clockwise(CCW) rotation

“clock_wise_step()” and **“counter_clock_wise_step()”** rotates the robot in the desired direction. Again in this case motion is planned in conservative way. In the CW & CCW legs are lifted and moved in the desired direction call function **“clock_wise_step ()”** for CW & **“anti_ clock_wise_step ()”** for CCW step. Once all the legs are moved, robot pushes itself in the desired direction by calling the function **“robot_stand_position ()”** Turning can be done faster as shown in the “Hexapod 2” video in the documentation CD.

6.8 Remote control

Functions described in this section are used in Hexapod_remote_control code.
In this application robot acts as master and remote act as slave. When robot wants data related to joystick position, it sends string of 4 characters as "NEXR", at 115200 bps to the remote via XBee wireless module. In return to this string remote sends header byte string as "FB" followed by 8 data bytes containing information of the joystick, Remote control's tilt, pan angel and digital switch position.

Transmitting packet from the robot:
N - 1st byte of packet
E - 2nd byte of packet
X - 3rd byte of packet
R - 4th byte of packet

Receiving packet to robot from remote:
F - 1st byte of packet (1st header byte)
B - 2nd byte of packet (2nd header byte)
Byte1 - 3rd byte of packet (Vertical position of left joystick)
Byte2 - 4th byte of packet (horizontal position of left joystick)
Byte3 - 5th byte of packet (Vertical position of right joystick)
Byte4 - 6th byte of packet (horizontal position of right joystick)
Byte5 - 7th byte of packet (Accelerometer x axis position)
Byte6 - 8th byte of packet (Accelerometer y axis position)
Byte7 - 9th byte of packet (Battery Voltage)
Byte8 - 10th byte of packet (out of 8 bit data bit0 shows right switch and bit1 shows left switch position and other bit are left for other user application)

The right joystick is used for robot motion control and left joystick is used for servo mounted camera pod control. Remote control also has built-in 2 axis accelerometer and switches which can be used for adding more functionality.

Communication is synchronized in the timer 1 ISR. In the demo code robot asks for the data form the remote control once every 200ms. Data update rate can be increased up to 10 ms by changing comparison value from 80 to 4 in the, if statement “if (serial_communication_interval > 80)”.

The remote replies the current joysticks, accelerometer, and switch position data in the packet form. The data packet is received in the UART0 ISR function **“SIGNAL**

(SIG_USART0_RECV)”. If the packet is valid then respective value is stored in the **“FBV_ctrl_data[]”** array and action is taken by calling the **“remote_control()”.**

After successfully receiving 10 such packets from the remote control, the on board buzzer beeps twice at powering on the robot.

If the tilt and pan servo camera pods are connected, they are controlled by left joysticks horizontal and vertical position of the remote control. The **“angle_tilt_servo_pod ()”** and **“angle_pan_servo_pod ()”** are used for there respective movement in the firmware.

Important

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 3.2 for battery charging.