



## **SASI INSTITUTE OF TECHNOLOGY & ENGINEERING**

(Approved by AICTE-New Delhi, Affiliated to JNTUK-Kakinada and SBTET-Hyderabad)

**TADEPALLIGUDEM - 534101, West Godavari District, Andhra Pradesh**

☎ : 08818-244986,987,989,990 Fax:08818-244628 Web : [www.sasi.ac.in](http://www.sasi.ac.in), [hodece@sasi.ac.in](mailto:hodece@sasi.ac.in)

**Department of Electronics & Communication Engineering**

**Academic Year 2014-15**

**Annexure I**

### **Project Title: Design of A Multi-Terrain Hexapod Robot**

#### **TABLE OF CONTENTS**

1.1	Abstract	2-2
1.2	Motivation	2-2
1.3	Objective	2-2
2.1	Block Diagram and Working	3-3
2.2	Technical specifications	3-10
2.3	Result and Analysis	10-11
3.0	Conclusion	11-11

### **1.1. Abstract**

In present days robotics is the vast field to explore things without human intervention. A robot is an electro mechanical device that can perform autonomous or preprogrammed tasks. Robots may be used to perform tasks that are too dangerous or difficult for humans to implement directly. This project mainly discusses the conceptual and detailed design and development of a Hexapod robot with minimum degree of freedom. Electronics architecture and control algorithms are briefly described. The project finishes with discussion of the current results and identifies some future works. This project is aimed to design and construct a robotic system which is used for military people. The configuration and characteristics of the mechanism are introduced. The robot mechanism, process of moving are analyzed. The principal theory which is used in the hexapod robot is elaborated, the control system of the robot is designed. Analysis shows that: The mechanism has the characteristics of compact body, easy control, and good move characteristics. DC servo motor helps the robot to move forward, backward, left, right, stop.

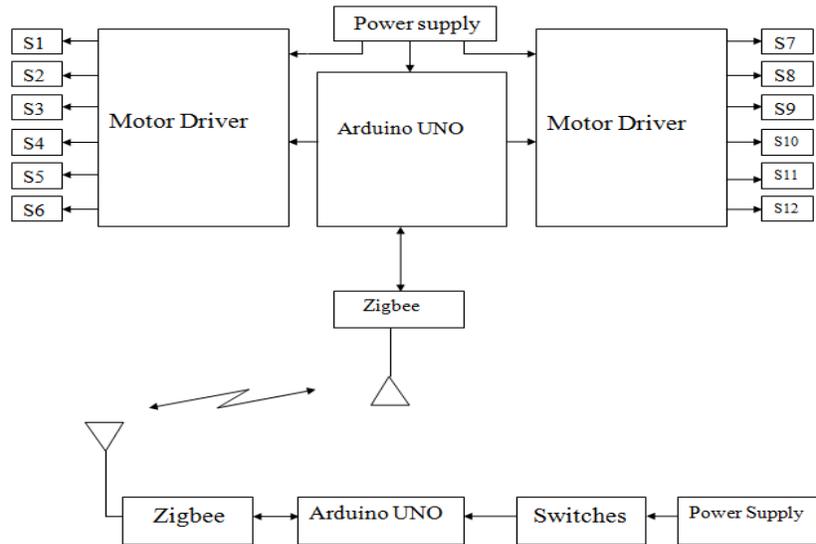
### **1.2. Motivation**

The robot is used for the military people for carrying the loads. If a military personnel wants to travel from one place to another place he must carry all his weapons, food, clothes etc. The person can place his luggage on robot and he can walk freely and also he can control the robot by himself.

### **1.3. Objective**

Now-a-days robotics is one of the most developing and advanced technology. Robotics is also useful to do jobs in areas and in situations that are hazardous for human. Our preliminary aim in this project is to build a robot which can carry luggage from a particular place to a new location. And we also had remote controller for the control of the robot. This project increases the speed of the people who are travelling in different types of terrains.

## 2.1. Block Diagram



**Figure 2.1: Block Diagram of Hexapod robot and the remote controller**

**Working:** The fig:2.1 shows the block diagram of a hexapod robot .It mainly consists of a microcontroller which acts as a controlling unit. Remote controller is used as an input module that controlled by the user to move the robot in desired direction. The instructions from the microcontroller will help to run the motors. Their will a wireless ZigBee transmission between a remote controller and robot.

## 2.2. Technical specifications

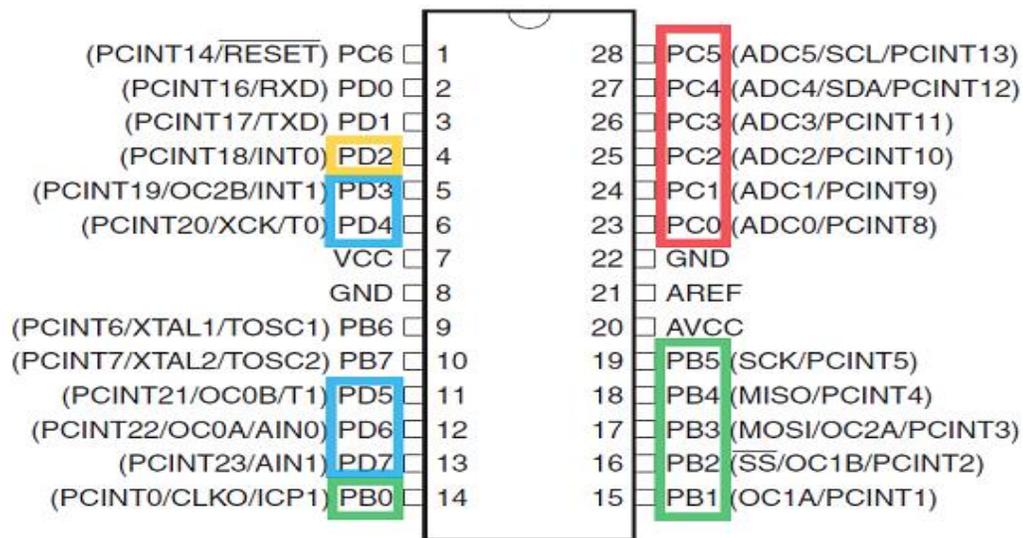
### 2.2.1 Arduino Uno Board

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

#### Features:

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V

- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by boot loader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz



**Figure 2.2: Pin diagram of ATmega328**

### *Arduino Technical Specifications*

- |  |   |
|--|---|
| • Operating Voltage                        | 5V  |
| • Supply Voltage (recommended)             | 7-12V   |
| • Maximum supply voltage (not recommended) | 20V   |
| • Digital I/O Pins                         | 14 (of which 6 provide PWM output)                    |
| • Analog Input Pins                        | 6   |
| • DC Current per I/O Pin                   | 40 mA   |
| • DC Current for 3.3V Pin                  | 50 mA   |
| • Flash Memory                             | 32 KB (ATmega328) of which 0.5 KB used by boot loader |
| • SRAM                                     | 2 KB (ATmega328)                                      |
| • EEPROM                                   | 1 KB (ATmega328)                                      |
| • Clock Speed                              | 16 MHz  |

### **2.2.2. Zigbee System**

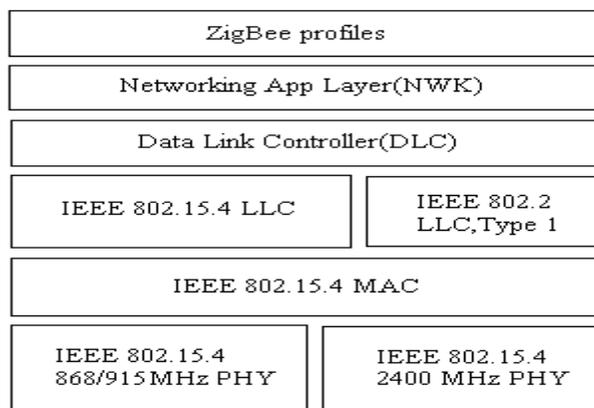
ZigBee is a home-area network designed specifically to replace the proliferation of individual remote controls. ZigBee was created to satisfy the market's need for a cost-effective, standards-based wireless network that supports low data rates, low power consumption, security, and reliability. To address this need, the ZigBee Alliance, an industry working group, is developing standardized application software on top of the IEEE 802.15.4 wireless standard. The alliance is working closely with the IEEE to ensure an integrated, complete, and interoperable network for the market. For example, the working group will provide interoperability certification testing of 802.15.4 systems that include the ZigBee software layer.

**Why do we need ZigBee technology?**

- Addresses the unique needs of most remote monitoring and control and sensory
- Network applications.
- Enables the broad-based deployment of wireless networks with low cost, low Power solutions.
- Provides the ability to run for years on inexpensive primary batteries for a typical monitoring application.
- ZigBee was realized that this technology is low cost.

**ZIGBEE Layer Architecture:**

The new ZigBee protocol provides an open standard for low-power wireless networking of monitoring and control devices. Working with the IEEE 802.15.4 standard which focuses on low-rate personal area networking and defines the lower protocol layers (i.e., the physical layer, or PHY, and the medium access control layer, or MAC)—ZigBee . ZigBee uses the IEEE 802.15.4 physical and MAC (medium access control) layers to provide standards-based, reliable wireless data transfer. ZigBee adds network structure, routing, and security (e.g., key management and authentication) to complete the communications suite.



**Figure 2.3: ZigBee Architecture.**

On top of this robust wireless engine, ZigBee profiles provide target applications with the interoperability and inter compatibility required to allow similar products from different manufacturers to work seamle defines the upper layers of the protocol stack, from network to application, including application profiles. Think of 802.15.4 as the physical radio and ZigBee as

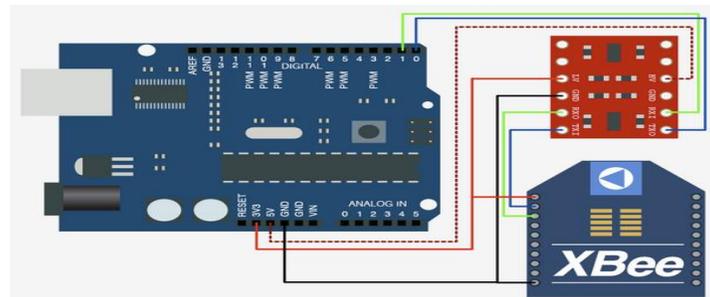
the logical network and application software ZigBee uses the license-free ISM bands, which provide unrestricted geographic use.

The new standard targets home and building control, automation, security, consumer electronics, PC peripherals, medical monitoring, and toys. These applications require a technology that offers long battery life, reliability, automatic or semiautomatic installation, the ability to easily add or remove network nodes, signals that can pass through walls and ceilings, and low system cost.

ZigBee and the underlying 802.15.4 standard offer the system designer several classes of devices: the reduced-functionality device (RFD), the full-functional device (FFD), and the network coordinator. All ZigBee networks have at least one RFD or FFD and a network coordinator. Most sensor applications fall natively into the RFD class, with extended networks making use of both FFDs and network coordinators to form bridges and links required by the network topology. ZigBee networks can form autonomously, based on connectivity and function.

### ***ZigBee Interfacing with Microcontroller***

The focus of network applications under the IEEE 802.15.4 ZigBee standard include the features of low power consumption, needed for only two major modes (Tx/Rx or Sleep), high density of nodes per network, low costs and simple implementation. The ZigBee has 20 pins. In our project the transmitter of the Arduino is connected to receiver of the ZigBee. The receiver of the Arduino is connected to the transmitter of the ZigBee. The communication between the ZigBee module and the Arduino UNO is done by cross legged connection.



**Figure 2.4: Interfacing Of ZigBee With Microcontroller**

### **2.2.3. Servo Motors**

#### ***Servomotor Driver***

A servo motor controller is a circuit that is used to control the position of a servo motor. It is also called as a servo motor driver. A servo motor controller consists of a controller, the servo motor and the power supply unit. Servo motor driver may be used to control a single servo or even a group of servo motors. In many projects where servo motor controlling is the mainstay of the task to be accomplished, the controller must drive more than one servo. An example of this is an RC airplane, which uses many servos.

### ***Essential-components***

1. Micro-controller
2. A power supply unit.

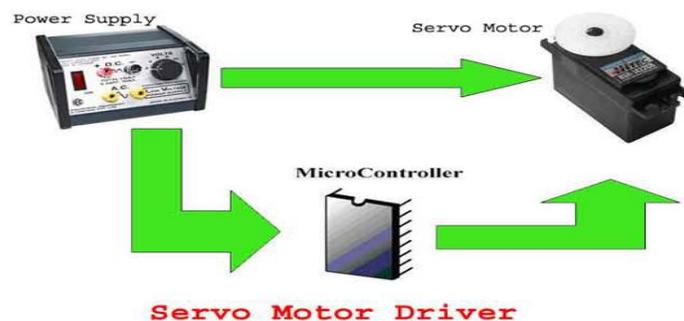
### ***Micro-controller***

A servo motor is driven by applying the voltage signal to it regular intervals. The servo is sensitive to timing variations. A pulse of specific width has to be applied at specific intervals of time. Typically, the duration of pulse varies from 0ms to 2.2ms and the repetition rate is 50Hz to 60Hz. For precise position control, the controller that is chosen must have timers that have the required resolution. Also, if more than one motor has to be controlled simultaneously, the processor clock must be fast enough. For a single motor control, an 8051 can be used like a AT89s51 or a P89v51RD2. But for more than one motor, we must use a PIC, like a PIC18F or a ATMEGA, so that it's internal PWM can be utilized. However, the selection of micro-controller depends totally on the designer and the project requirements.

### ***Power Supply***

The design of the power supply unit servo motor controller depends on the number of servo motors that are interfaced to the board. Servo motors operate from 4.8V to a 6V supply voltage. The typical value is 5v. Applying voltages greater than the supply voltage is not advisable as it may render the motor permanently useless. The current draw of the motor is variable and depends on the torque that it generates. Also it will draw less current when in idle mode and more current when it is running. A servo motors maximum current draw is given as its stall current. This is the maximum current it will draw when running with the maximum torque before it stops due to overload. This current value can be as high as 1 A for some motors. For a single motor control a voltage regulator like a LM317 can be used along with a suitable heat sink. But when multiple motors need to be interfaced, a high quality supply with higher current rating must be used. A SMPS (Switched mode power supply) can be a good option.

Block Diagram below showing interconnections in a Servo Motor Driver

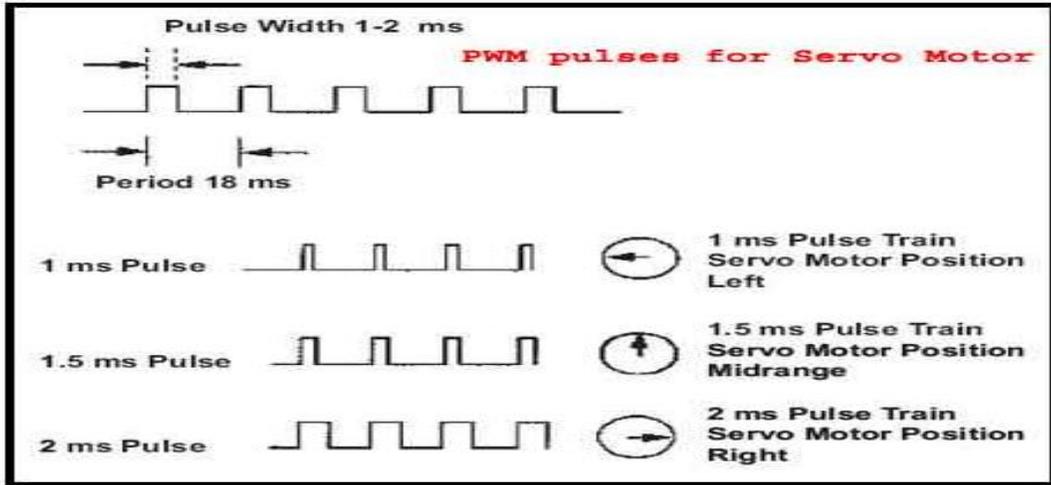


**Figure 2.5: Interconnections in a Servomotor Driver**

### ***Controlling Servo Motor***

The servo motor has three terminals. Position signal (PWM Pulses),  $V_{cc}$  (From Power Supply), Ground. The servo motor angular position is controlled by applying PWM pulses of specific width. The duration of pulse varies from about 0.5 ms for 0 degree rotation to 2.2 ms for 180 degree rotation. The pulses need to be given at frequencies of about 50Hz to 60Hz.

In order to generate the PWM (Pulse Width Modulation) waveform, as shown in figure below, one can use either the internal PWM module of the micro-controller or the timers can be used. Using the PWM block is more flexible as most micro-controller families design the blocks to suit the needs of application like Servo motor. For different widths of PWM pulses, we need to program the internal registers accordingly. Now, we also need to tell the microcontroller how much it has to rotate. For this purpose, we can use a simple potentiometer and use an ADC to get the rotation angle or for more complex applications an accelerometer can be used. The below fig:2.6 shows PWM pulses for servo motors.



**Figure 2.6: PWM pulses for servomotor**

***Towerpro Sg90 Micro Servomotor 9g***

The fig: 2.7 tower pro servomotor which we used in this project and its dimensions. Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware. fig:2.7 shows the output waveforms of servomotor.



**Figure 2.7: Towerpro Sg90 Micro Servomotor 9g**

Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "180" (~1 ms pulse).

### ***Specifications***

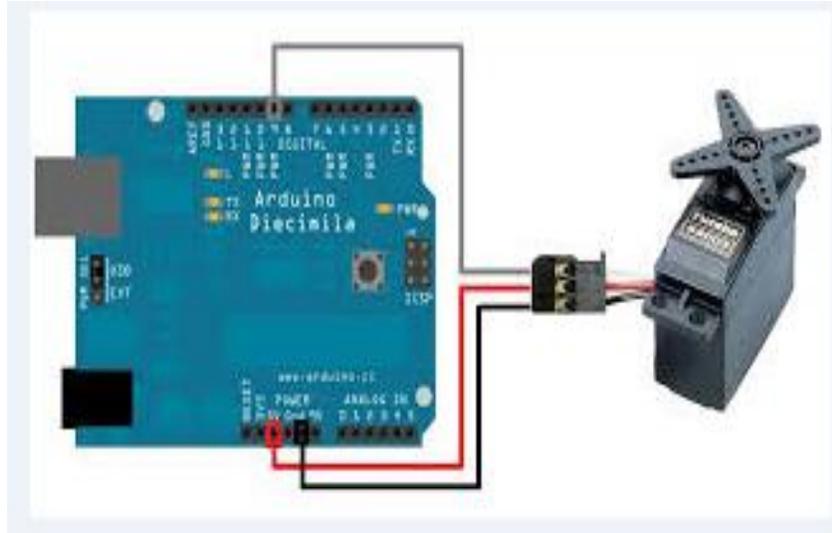
- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10  $\mu$ s
- Temperature range: 0 °C – 55 °

### ***Interfacing of Servomotor With Arduino Uno***

Servo motors are small controllable motors that lend to implementation in many applications. There are Servos with many different speeds, sizes and torque capabilities, but all have 3 wires, power, ground and control. Servo motors are popular with hobbyists because they are inexpensive, \$15-\$100, and control of Servo motors with microcontrollers is universal for all models. Servo's receive pulse width modulated (PWM) signals to determine in which manner to move. There are many ways to send this signal to the motor; this application covers how to send the desired PWM signal to the Servo motor using the Arduino UNO microcontroller.

The Arduino UNO is a popular microcontroller which comes on a development board to accelerate programming, provides simple interfacing with peripheral devices and connection with computers. The Arduino UNO chip is programmed with the Arduino programming language and Arduino 1.0 software through a USB port on the board which plugs into a computer's USB port. The Arduino 1.0 software writes code from the software to the chip by uploading the file containing the desired code to the board. Once the chip has been programmed with the desired code, the chip can be removed from the development board and connected to any circuit. There is much literature on this board and in the reference section of this document there are several places to find more information on this topic. Specifications Servo motors are controlled through the control line, usually a yellow or white wire.

The pulse width of the signal sent to the Servo control wire determines how the motor will move, either clockwise or counter clockwise. When the Pulse Width is less than 1.5ms the motor will move to the 0 positions and hold. When the Pulse Width is 1.5ms the motor will rotate to the 90 degree position and if the Pulse Width is greater than 1.5 ms the motor will rotate to the 180 position. When the motor reaches the desired position it will hold there until a signal is sent to move. This is done in this application using the Arduino 1.0 coding software to write to one of the Arduino UNO's 5 PWM output pins. The PWM output pins on the development board can be written to with different pulse widths which are used to control the motor. The fig:2.8 shows the interfacing of microcontroller with servomotors.



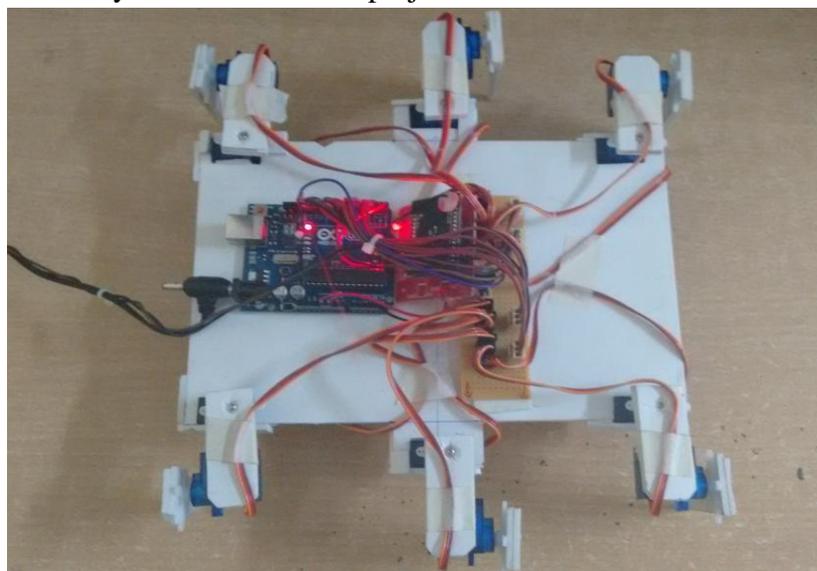
**Figure 2.8: Interfacing Of Microcontroller with Servomotors**

#### **2.2.4. Power supply:**

With just two parts your Arduino goes portable, powered by a 9V battery. Note that these connections are the same ones you'd use to make a plug to connect to a solar cell array, a car lighter plug or a larger battery. It needs a 9V battery clip (snap connector), a 2.1mm power plug, a soldering iron and some solder, and optionally a small piece of heat shrinks tubing. Any hobby shop should have these items. First, slide the outer shell of the plug over the battery clip's wires. Optionally, slide a piece of heat-shrink tubing over the red (+) wire to protect the positive connection. Next, solder the battery clip's black (-) wire to the outside connection of the plug. Then solder the battery clip's red (+) wire to the center connection of the plug.

#### **2.3. Results and Analysis**

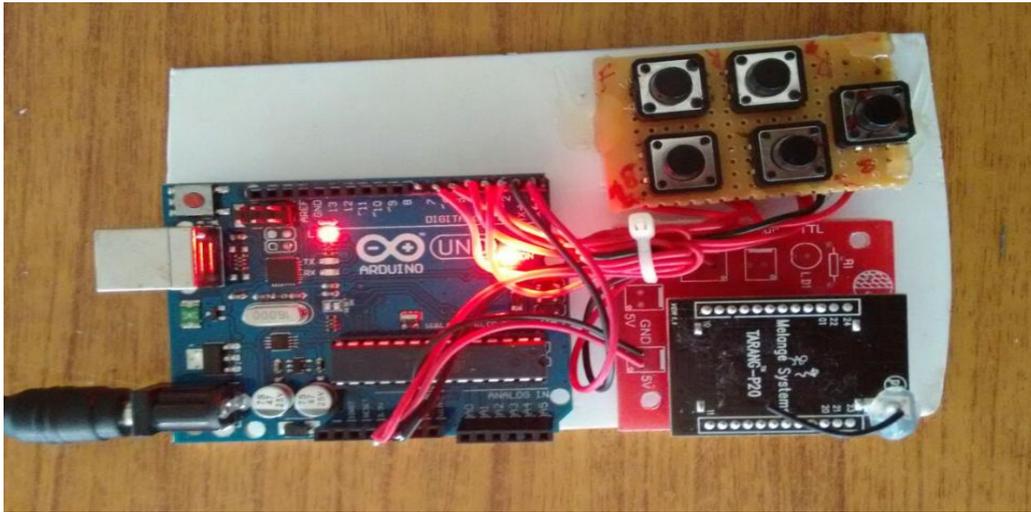
The different applications of this hexapod robot, pictures and description of our project has been made and finally conclusion of our project.



**Figure 2.9: Hexapod Robot**

The above fig:2.9 represents the hexapod robot. We use Arduino Uno microcontroller, Tower pro micro servo 9g-SG90, and a ZigBee module for the communication. For controller the robot we use a remote controller. The remote controller consists of Arduino Uno board and a ZigBee module. fig:2.10 shows the remote controller of hexapod robot.

There will be the wireless communication between remote controller and the robot through ZigBee up to 150 m. The connection between the Arduino board and the ZigBee module is the cross connection. The transmitter of the ZigBee is connected to receiver of the Arduino and the transmitter of the Arduino is connected to the receiver of the ZigBee. The 12 servomotors are connected to the digital pins of the microcontroller. The remote controller consists of five switches. They are forward, Backward, left and right. And the stop switches for stopping the robot. The fig:2.10 shows the remote controller.



**Figure 2.10: Hexapod Robot Remote Controller**

### **3.0. Conclusion**

The growth of science and technology makes human life easier. The main theme of this project is to build a hexapod robot, which can be used for military purpose. This designed architecture is suitable for all terrains like rigid and smooth surfaces. A hexapod robot can take an advantage of multiple legs to walk easily. In this design the robot is widely useful in the military purpose, where the soldier usually carry the weapons which are heavy. The weapons can be loaded on to the robot and carried along by the troop. The control of robot can be wirelessly done by means of a ZigBee module.

The project “Design of a Multi Terrain Hexapod Robot” has designed and tested. Integrating features of all the hardware components and have developed it. Presence of every module has been reasoned and placed carefully thus contributing to the best working of the unit. Secondly, used advanced hardware and with the help of innovating technology the project has been successfully implemented.