WICSER ROBOT
(A WIRELESSLY COMMUNICATING SELF- RECONFIGURABLE ROBOT)

Project Advisor:

Dr. Ali Raza

Authors:

Muhammad Fahad Osmani 2009-MC-07

Sundas Zahra 2009-MC-08

Mamona Awan 2009-MC-30

Department of Mechatronics and Control Engineering,

University of Engineering and Technology, Lahore.

WICSER ROBOT

THIS PROJECT IS TO BE SUBMITTED FOR THE PARTIAL FULFILMENT OF

BACHELORS DEGREE

IN

MECHATRONICS AND CONTROL ENGINEERING

APPROVED BY

INTERNAL EXAMINER	__________________	**EXTERNAL EXAMINER**	__________________
SIGNATURE	__________________	**SIGNATURE**	__________________

DEPARTMENT OF MECHATRONICS AND CONTROL ENGINEERING

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, LAHORE

Acknowledgements

Thanks to *Allah Almighty* who gave us the strength and courage to work on this project and achieve our desired goal with His grace and blessings.

We would like to thank our project advisor *Dr. Ali Raza* under whose supervision we completed our project. His valuable suggestions, guidance, continuous support and motivation were our greatest asset throughout our project work and are viewed by us as a major factor helping us in the completion of the project.

Also, we extend our heartiest gratitude to all the faculty members of our department who were present at all the project presentations, giving us valuable suggestions regarding our project which helped us throughout our work.

Finally, we offer regards and prayers for all the technicians, lab personnel and our friends who helped and supported us in this project.

Special thanks to Mr. Muhammad Munawar un Noor from his wife for supporting and encouraging her at every step and motivating her to complete the graduation.

Dedication

This project is dedicated to

our beloved parents

whose prayers, affection and tireless efforts

made us what we are today

and gave us the strength and courage

to work for the realization of our goals

and reach this greatest point

of achievement.

Abstract

ROBOTICS is the key to modern human evolution. Throughout the course of history, mankind has evolved significantly from being ‘cave men’ to the most sophisticated ‘social animals’ in the world with the revolutionary technological innovations serving as the foundation for this profound evolution which has changed and will continue to change the face of the world we know.

Minimizing human intervention and increasing the role of machines in every field of life is the major emphasis of modern engineers, increased efficiency being the desired objective. This technological progress is not only restricted to the large scale space explorations but is also finding increasing application in our domestic circle.

Keeping in view the modern technological progress and the worlds’ inclination towards robotics, we worked on the prototype of a **self-reconfigurable robot** designed to work in the environments that are not easily accessible by human beings.

Self-Reconfigurable Robot is a modular robot that is capable of reconfiguring itself to adapt to the changing environment. Reconfiguration can be in terms of shape or functions according to the changing environment and demands. Project aims to design the modules of the robot, synchronize them to execute appropriate motions through genetic algorithms. Modules are capable of rearranging their mutual mechanical connections to change the robot’s outward features. Overall the robot realizes **reliability, quick self-reconfiguration and versatile robotic motion.**

This project report contains the mechanical design, the electrical design and programming algorithms of our project. The design and fabrication process is explained in detail. The associated problems and their solutions have been described. The report is divided into various chapters.

All references and relevant extracts from the datasheets have been provided in the appendices at the end of the report.

Contents

List of Figures

Chapter 1: Introduction

1.1) Introduction

The desired objective of our project is to build a SELF-RECONFIGURABLE that can change its shape and functionality without external intervention. Such robots are composed of many similar robotic modules which are equipped with computational and communication capability, sensors and actuators. The component modules change connective relations among themselves. Self-reconfigurability means that the system can metamorphose into various shapes by changing connectivity among the modules and gait programming. Many configurations, such as a manipulator, a crawler, a legged robot or other robotic configurations, can be built by the combination of (identical) modules. This kind of flexibility is highly desirable for robotic systems used in unstructured and unpredictable environments.

Thus following hypothesis can be developed on the basis of present discussion

"This reconfigurable robot will be more efficient as compared to its previous versions as it will use wireless means of communication i.e. RF MODULES"

This project has the following sub-sections:

1. Modules
    i. Active block
    ii. Passive block
2. Micro-Controller
3. Power sources
4. Connecting Mechanism
5. Communication Device
6. Motors

These sub-sections would be explained in full detail in the coming chapters.

1.1.1) Project Description

A self-reconfigurable robot is comprised of various robotic modules. Modules can be connected in many ways. Motion of modules has relationship with each other to change the whole robot shape. This shape shifting makes it possible for the robots to adapt and optimize their shapes for different tasks. Association of robotic modules requires impeccable communication between them in order to alter their assemblage.

Self-reconfiguration will be practiced by the incorporation of numerous modules. A module is envisioned to be self-govern to change its directions and form. Mechanical structure will be first designed under the consideration of required dimensions and to meet out the desired constraints. Most studies have focused on a distributed algorithm or a planning method for structural formation. As for locomotion, project is whole body locomotive. This is realized by controlling joint motors, without any configuration change. This motion is suitable for faster motion in case of a simple configuration such as a 4-legged or a snake-like robot.

Module comprises two blocks and a link. Both blocks have identical shape, comprising a half-cube and a half-cylinder. Two blocks can rotate independently about their axes by ±90 deg. At any angle, their half-cylindrical parts always have mutual physical contact. The two blocks differ in gender: opposite gender blocks of two modules can connect mechanically with a counterpart by their three flat surfaces. Although the three surfaces shapes are not identical, they can all connect symmetrically in four possible orientations. Using this flexibility of connection, various structures that are suitable for locomotion can be constructed as a chain-type modular robot. For self-reconfiguration, a module must fits in two cubic blocks so that mechanisms, circuits, and wires are all installed inside the three block parts. The passive (female) block contains a power circuit board, a battery and a controller board. The link block contains two geared-motors and their controller circuit. The active (male) block contains three mechanisms for surface connection and their controller circuit. On the three sides of a block there are permanent magnets working along with SMA coils that are assembled for the attach and detach of the modules.

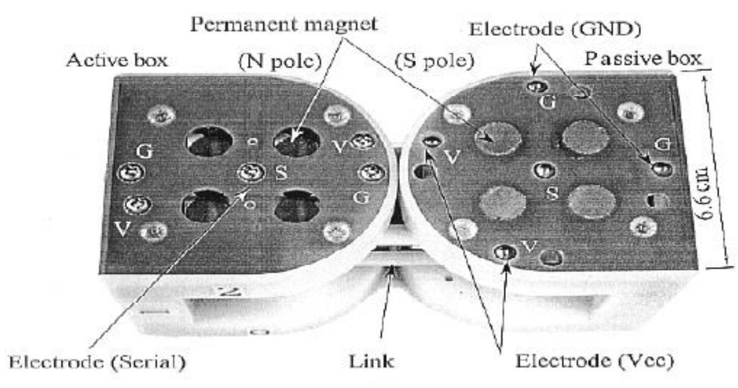


Figure 1 Desired Module

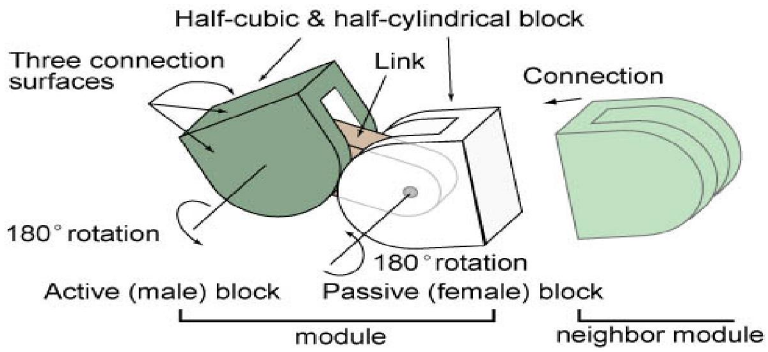


Figure 2 Desired Module description

1.1.2) Modification

Instead of using electrodes for serial inter-module communication, our project will utilize RF (radio frequency) Modules for **wireless communication.** Furthermore a third RC servo motor will be incorporated for axial rotation within a single module.

Figure 3 Morphology-I

As shown in pictures module have been joined in two different configurations for different situation handling.

Figure 4 Morphology-II

1.2) The Potential of Self-Reconfigurable Robot

By having the ability to transform themselves into different shapes, Self-Reconfigurable robots have the potential to exceed their conventional counterparts in multi-functionality[1], flexibility, and robustness.

1.2.1) Morphogenesis (Self-Assembly)

Traditional robots have a fixed configuration. For example, if a robot is designed to be two-legged, it stays two-legged forever. However, Self-Reconfigurable robots actively change their configuration to achieve their goals. This process is called morphogenesis, or shape-shifting .Typical morphogenesis include finding an algorithm to shift from any shape to a specific one, finding a reconfiguration path from one certain configuration to another; and finding a suitable configuration under the given environment.

1.2.2) Self-Reproduction

A system's ability to produce a copy of itself is called self-reproduction. Because of its modularity, self-reproduction is much easier for a Self-Reconfigurable robot than for an ordinary robot.

1.2.3) Scalability

When a system's functionality depends linearly on the number of modules, the system is called scalable. However, it is only concerned with partial malfunction. Scalability, one of the basic properties of an information network, is difficult to realize in a mechanical system. However, in a Self-Reconfigurable robot it is relatively easy.

1.2.4) Motion Generation

The major task for a self-reconfigurable robot is the motion generation. It is made easy through genetic algorithm.

1.2.5) Co-Evolution of Shape and Motion

The shape of a robot constrains its motion and the motion affects its shape. Shape and motion are intrinsically indivisible, and they actually are two different aspects of the evolutionary cycle. With Self-Reconfigurable robots, we are able to examine the same process in the real world by hardware, not in the virtual world of computer graphics.

1.3) Motivation

Robotics is the key to modern human evolution. Throughout the course of history, mankind has evolved significantly from being ‘cave men’ to the most sophisticated ‘social animals’ in the world with the revolutionary technological innovations serving as the foundation for this profound evolution which has changed and will continue to change the face of the world we know.

Keeping in view the modern technological progress and the worlds’ inclination towards automation, we worked on the prototype of an ‘**Automated Library System’** designed to retrieve the books automatically for the clients.

‘Self-Reconfigurable Robot’ involves a rigorous mechanical and electrical design. It also involves an ample use of computer programming as the micro-controller is programmed to control the various sub modules. Also, this project involves the communication intelligence through RF-MODULES.

Therefore, this project encompasses all three major areas of **Mechatronics,** namely Mechanical, Electrical and Computer Programming and hence we believe it was extremely suitable to be chosen as a final year project.

Moreover, the design and fabrication of the WICSER ROBOT covers a variety of the important subjects of Mechatronics Engineering and therefore gives a wide learning spectrum by the practical implementation of the theoretical knowledge. These subjects are as follows

- Embedded Systems Programming
- Computer Interfacing and Networking
- Automation
- Instrumentation (Sensor and Actuators)
- Machine Design, AUTOCAD and COREL
- Electronic Devices and Circuits

Chapter 2: Background

2.1) Literature Review

SuperBot is being designed for NASA space exploration programs. Similar to M-TRAN, a network of SuperBot modules can perform as both lattice-based and chain-type self reconfigurable robots. Its network is capable of transmitting data at high rate using infra red LEDs[2].Since sensory information is in distributed form due to distributed nature of network, so this information is to be accumulated in order to make a right and purposeful decision. It may be possible that only some of the modules use their power at a time, this may cause their battery to drain out; so modules must have a source of sharing power through the network. M-TRAN does not possess the ability to roll like SuperBot, however it can be done by a series of bending, docking and undocking sequence which will cause more power consumption. Hence SuperBot has some particular advantages over M-TRAN modular robot.

This article states about the scheme for underwater navigation of self-reconfigurable robots. This is achieved by re-arrangement of modules and neural controllers for locomotion. Neural controllers prove to be promising for such robots due to their distributed nature and ability to generate locomotion for complex (multi degree of freedom) structures[3].

This article presents two evolutionary motion generation methods using generic algorithms for modular robots. In first approach motion sequence is described as a series of segments specifying dynamic motion. Using the travelling distance as fitness function, full body dynamics were evolved[4]. Second approach concerned Central Pattern Generator (CPG's) for fixed configurations.

Self-reconfigurable robots can be controlled simply by the use of several generic locomotion algorithms[5]. Reconfiguration algorithms have been developed for all of these hardware systems, predominately centralized, but numerous distributed control algorithms have been proposed. These algorithms allow a group of modules to roam over place, to build tall structures, to overcome obstacles and enter enclosed spaces. They manage to keep the particular activity precise and rapid. These types of algorithm are now extensively used for the control of such robots.

In this article hybrid communication system for modular robots is discussed. This type of communication system has an impressive ability to connect on demand to form different topologies. Communication is central to modular robots. Local type is to figure out topologies and global type is needed for tasks which require some combine assessment. According to the research and the experiments performed Hybrid type of communication provides best approach for modular robots[6]. They improve transfer rates, flexibility, space, power conservation, efficiency and reliability.

The self reconfigurable robots assure three features versatility, robustness and low cost. The PolyBot systems have proved that multi module systems can be very versatile using different modes of locomotion. Programming self collision free locomotion is a difficult job as it has many modules in chain and size of this workspace is exponential in the number of modules. It has been verified that gait control tables were effectively able to control the gait of the robot[7].

Multimode locomotion is a essential feature for any self reconfiguring robotic system to sustain. In different environments such robots must use different modes of locomotion. To support multimode locomotion, a robot must be able to perform different locomotion strategies, recover from unexpected locomotion failures, shift from one mode to another and choose the correct mode for the correct environment[8]. This paper elaborates a case study of multimode locomotion with a reconfigurable modular robot called SuperBot. SuperBot modules can support many different locomotion modes (such as rolling track, snake, caterpillar, insect, spider, H-walker, etc.). Each SuperBot module has three joints; the middle joint can mechanically rotate continuously in both directions, the other two are same as M-TRAN modules. Thus application of different locomotion modes without increasing the hardware and software complexity of the robots is possible.

Paper focuses on the development of such generic locomotion algorithms using geometric rules to control module actions. These generic locomotion algorithms can be applied to any modular robotic system e.g. Meta modules, Hexagonal lattice system and even MTRAN. The concept is extracted from cellular automation technique[9] i.e. by apply a set of local rules on the neighboring cell followed by actuation, the cell is moved to a new configuration. Three different types of actuation models are also discussed i.e. D0: the cells are evaluated in a set cyclic order so that there is variation in the relative delay between any two cells , D1: a cell can delay actuation at most one cycle relative to any other cell and D-infinite: any arbitrary delay between actuation. The set of local rules ensure locomotion without any deadlock or disconnection.

Extracting the concept from bio-mimetic research field ,the modular robots under discussion used neural oscillators (just like human nervous system located in spinal-cord for producing locomotion) as the controller of the joint motors to produce various locomotion patterns where the phase difference between the joint angles determined through connected CPG’s (central pattern generators) . In neural oscillators each neuron is represented by a non-linear differential equation. Each module’s motor has its own CPG. The performance of the locomotion pattern is evaluated using fitness function as given below

$Fitness = (a * length) - (b * width) - (c * (loss / num))$ [10][11]

Here a, b and c are weighting co-efficient. Length is the moving distance of centre of gravity in the positive z-axis, width is the moving distance in x-axis, loss is the energy loss and num is the number of modules. An efficient locomotion pattern is generated by using genetic algorithms by i.e. each individual is evaluated by fitness and after selection, mutation and cross-over the optimal pattern is evolved.

The paper discusses the two modular robots i.e. MTRAN-1 and MTRAN-2, with the detail Electrical and mechanical design, communication mechanism. Various improvements of MTRAN-1 that are implemented in MTRAN-2 are discussed e.g downsizing from 66mm to 60 mm and 440g to 400 g weight, more reliable communication via Global communication Lon Works & RS-485 (39kbps) and Local communication Asynchronous serial 4,800 bps[12]. In MTRAN-2 there are two layers of multi-CPU and inter-module network system, which has upgraded the performance of the control system.

Mechanical design and hardware of M-TRAN and software for self-reconfiguration and motion generation was explained in this paper. Derlin plastic was used to make the module body, Permanent magnets and shape memory alloys for attachment and detachment of modules[13]. For electrical connection, two pairs of electrodes for power supply and one electrode for serial communication are also embedded in the surface.

The passive box contained the processor and other circuitry. The circuitry included a microprocessor a power supply circuit and a pulse generator circuit for driving the SMA coils. The microprocessor had a dipswitch input to set an ID number. Communication was asynchronous serial communication. The host PC and the relay PIC were connected by bilateral communication, while single-wire communication with token passing was adopted between the relay PIC.

The host PC was giving a control command that included the module's ID number. The relay PIC then transmitted the command to all the modules on the serial bus. A module sent back a proof signal if its ID coincided and executed the command[14]. After the execution of the command, it sent a completion signal to the host PC via the relay PIC. When no validation was returned from the module, the host PC sent the command again for recovery.

Self-reconfiguration (SR) was achieved by basic operations such as detaching a surface from the neighbor, rotating a semi cylindrical box and reconnecting the surface to another module.

MTran did not have many degrees of freedom (DOFs) but a module had enough torque to lift one module in any posture. Due to less DOF straight and forward motion planning was not possible. Two types of motion planning were considered. One was global planning and other was local planning. Global planner decided the overall motion called flow for the cluster to trace the 3D trajectory that what configuration had to be achieved. Local Planner divided the global planner into single ones, based on local rules. Each rule in a list includes a motion scheme that was local step motion. From a set matched rules the local planner selected one rule that had given any motion along the path given by global planner.

This paper described that a hormone inspired control structure for adaptive communication and distributed control in self-reconfigurable robots. This paper describes the adaptive communication (AC) protocol that permits modules continuously to make changes in their local topology, and the adaptive distributed control (ADC) protocol that allows modules to use hormone-like messages in their actions to accomplish locomotion and self-reconfiguration[15].

In adaptive communication, each module in a dynamic network, as an active cell, could continuously discover its local topological changed and adjusted its communication policy accordingly. They designed the adaptive communication (AC) protocol for all modules to discover and monitor their local topology and ensured the correct transmission of hormone messages in the network. This property holds regardless of the changes in the network topology.

This paper was about Yamor a reconfigurable robot. Yamor named SRR had locomotive capabilities without redesigning and reshaping under the control by the Bluetooth technology. Bluetooth technology truncated not only the physical touch of human but also reduced the communication wires between the modules. Bluetooth technology had numerous advantages upon communication through wires. Bluetooth provides 2.4GHz licensed free bandwidth and it could communicate with other devices like PCs, mobile phones and laptops etc. Bluetooth technology reduced the development cost and offered new opportunities in terms of flexibility and robustness[16].

2.2) Evolution and Comparison of WICSER with Other Reconfigurable robots

2.2.1) MTRAN

2.2.1.1) MTRAN-1

It has two semi-cylindrical parts and a link. The modules can execute a -90 to 90 degree rotation independently through a geared motor embedded in the link. Derlin plastic was used to make the module body, Permanent magnets and shape memory alloys for attachment and detachment of modules17. For electrical connection, two pairs of electrodes for power supply and one electrode for serial communication are also embedded in the surface. The passive box contained the processor and other circuitry. The circuitry included a microprocessor a power supply circuit and a pulse generator circuit for driving the SMA coils. The microprocessor had a dipswitch input to set an ID number. Communication was asynchronous serial communication. The host PC and the relay PIC were connected by bilateral communication, while single-wire communication with token passing was adopted between the relay PIC. The host PC was giving a control command that included the module’s ID number. The relay PIC then transmitted the command to all the modules on the serial bus. A module sent back a proof signal if it’s ID coincided and executed the command18. After the execution of the command, it sent a completion signal to the host PC via the relay PIC. When no validation was returned from the module, the host PC sent the command again for recovery. An efficient locomotion pattern, realized by a proper locomotion controller, is automatically generated by genetic algorithm (GA). The GA optimizes the locomotion controller by evaluating the performance of locomotion by dynamic simulations.

Extension:

MTRAN-1 was improved to form Mtran-2 with the same basic module architecture.

2.2.1.2) MTRAN-2:

MTRAN-2 is much more improved as compared to the MTRAN-1 especially in terms of CPU processing speed, communication system, motor control, maximum torque and speed, power supply using a battery, power consumption, and size and weight .Downsizing from 66mm to 60 mm and 440g to 400 g weight, more reliable communication via Global communication LonWorks & RS-485 (39kbps) and Local communication Asynchronous serial 4,800 bps19. In MTRAN-2 there are two layers of multi-CPU and inter-module network system, which has upgraded the performance of the control system. Different methods are used in the two models, MTRAN 1 and 2, to heat the SMA coils. In M-TRAN 1, electric current is applied directly through the SMA coils but as the electric resistance of the coil is too small, much power is consumed by the driving circuits. This problem was over-come in M-TRA 2, by implementing miniature light bulbs inside the SMA coils to heat them. The power needed for detachment is 36 W (0.5 A current with 8V voltage), which is far less than the power needed in M-TRAN 1. Mtran-2 has two ways of power supplying i.e. among host computer and modules through tethers and power supply circuitry that converts the power to 8V and 5V and autonomously each module has a lithium-ion (Li-ion) rechargeable battery (3.8V, 700mAh).

Extension:

MTRAN-2 was further improved to MTRAN-3.

2.2.1.3) MTRAN-3:

It was a hybrid type self-reconfigurable system. Each module is two cube size (65 mm side), and has 2 rotational DOF and 6 flat surfaces for connection. It is the 3rd M-TRAN prototypes. Compared with the former (M-TRAN II), speed and reliability of connection is largely improved. As a chain type system, locomotion by CPG (Central Pattern Generator) controller in various shapes has been demonstrated by M-TRAN II.20

2.2.2) SuperBot:

The SuperBot modules fall into the hybrid architecture. The modules have three degrees of freedom each. The design is based on previous system MTRAN. Each module can connect to another module through one of its six dock connectors. They can communicate and share power through their dock connectors. Several locomotion gaits have been developed for different arrangements of modules. These modules communicate in such a way that they do not require unique ID's.21

2.2.3) Molecubes:

Molecube is another chain self-reconfigurable robot. In which each module is 0.65 kg designed in a cube with 100mm long edges and one rotational degree of freedom. Molecube's blocks possess their axis of rotation lined up with blocks longest diagonal. Communication between cubes of revision 1 is done serially from RS232 and electrodes likewise in Mtran-I, Mtran-II, Mtran-III. But cubes from Molecube revision 2 communicate by Bluetooth modules. One servo motor and two processor are used for the precise and controlled movement. The demonstration of self-sized machines has been demonstrated by the Molecubes.

2.2.4) Polybot:

Polybot is another chain type self-reconfigurable robot. This robot contains modules that are cubic shaped about 50mm and has one degree of freedom rotational. The modules have brushless flat motors with harmonic drive transmissions, free torque sensors, whisker touch sensors infrared proximity sensor, a low resolution CMOS camera. For attachment and detachment they use hermaphroditic connectors and shape alloy memory alloy actuated latches respectively. In Polybot the connecting plates are provided with permanent magnets for attachment, SMA coils for detachment, two photo diodes and four LEDs ensuring the attachment and detachment through 6 connecting sides of modules. PolyBot modular robot family that has demonstrated many modes of locomotion including: walking: biped, 14 legged, slinky-like; snake-like: concertina in a gopher hole, inch-worm gaits, rectilinear undulation, and side winding gaits; rolling like a tread at up to 1.6 m/s; riding a tricycle; and climbing: stairs, poles, pipes, ramps, etc.[22]

	Class, DOF	Author	Year	Communication	Power sharing	Controller	Docking sides/ module	camera	Proximity sensor
CONRO	chain, 2 3D	Will & Shen (USC/ISI)	1998	Adaptive Communication-protocol	yes	PIC	6	no	no
PolyBot	chain, 1 3D	Yim et al. (PARC)	1998	Controller Area Network bus	yes	PIC	2	CMOS camera	yes
M-TRAN I	lattice, 2 3D	Murata et al.(AIST)	1999	serial	yes	PIC	6	no	no
M-TRAN II	lattice, 2 3D	Murata et al., (AIST)	2002	serial	yes	PIC	6	no	no
Superbot	chain, 3 3D	Shen et al., (USC/ISI)	2004	Adaptive Communication-protocol	yes	PIC	6	no	no
M-TRAN III	chain, 2 3D	Kurokawa et al., (AIST)	2005	Seria	yes	PIC	6	no	no
Molecubes	Chain,1 3D	Zykov, Mytilinaios, Lipson (Cornell)	2005	serial	yes	PIC	4	no	no
Project **(WISCER ROBOT)**	Chain, 2,2D			RF-module	No	Arduino	4	no	yes

Comparison:

Re-configurable robot (WICSER Robot) constructed will have the structural resemblance with MTRAN. Just like PolyBot, the robot will have a proximity sensor to sense and undergo an automatic attachment with other modules. Communication mechanism is completely revolutionized and RF module will be used to serve the purpose. Magnetic ad-hocking surfaces will be less than the MTRAN to make it cost effective. Arduino UNO will be used as Micro-controller which is never used before in any re-configurable robot.

Chapter 3: Methodology

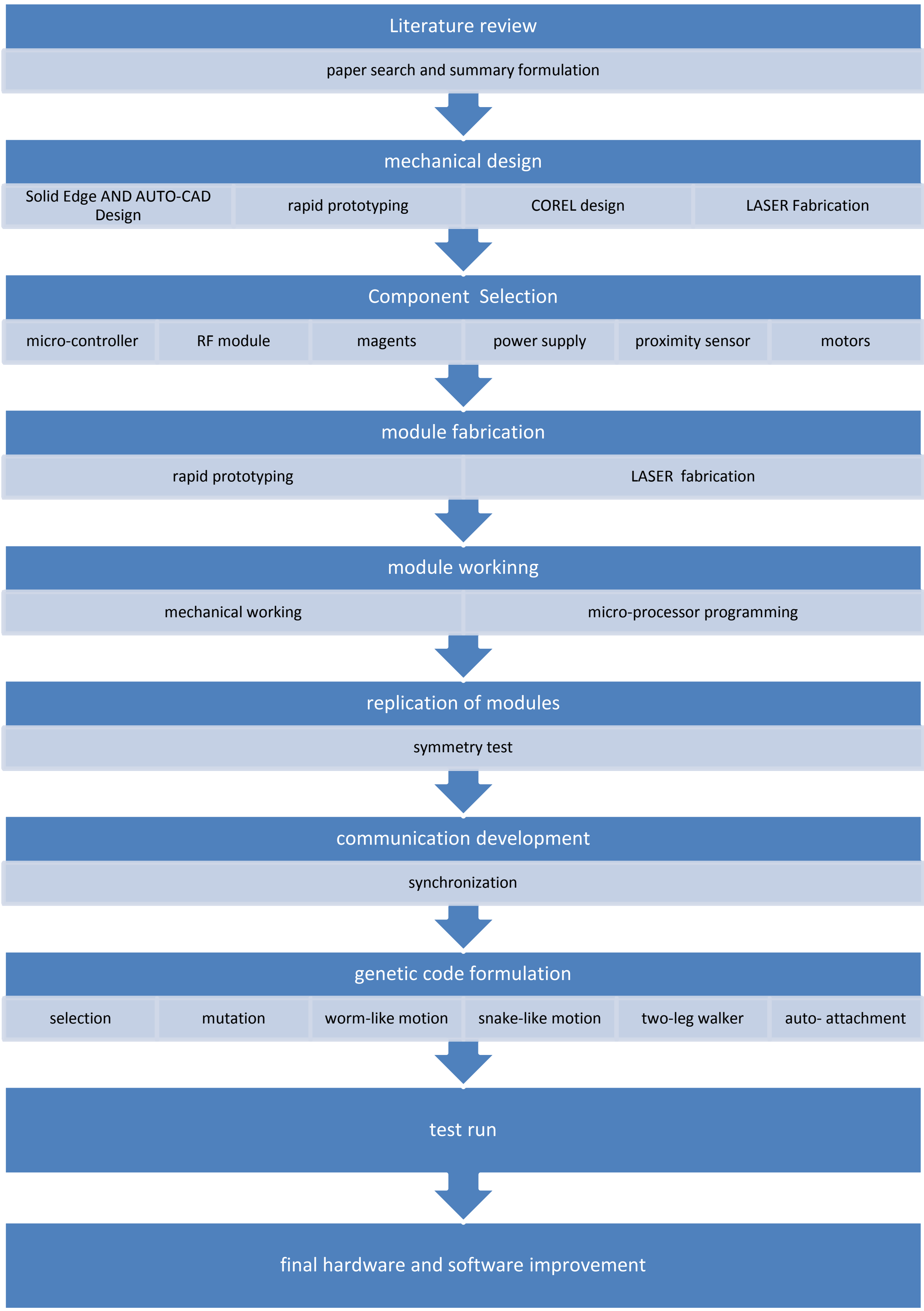
Methodology

In the first step, literature regarding modular robots will be thoroughly studied. Information related to design (mechanical and electrical), programming and communication will be gathered, summarized and a paper will be formulated.

Next step comprises of the designing of the module prototype. Critical analysis will be done to select the most suitable material for the modules to be most efficient. Then there will be the designing of the module. SOLID EDGE design or in other case AUTO-CAD design will be developed and this will be fabricated through rapid prototyping. Or in other case design will be developed in COREL. It will be fabricated on LASER machine and molded on flame.

Its mechanical and electric circuitry design will be finalized and fabricated. Proper selection of the components and order will be placed. Then the module testing will be carried out to check the working of the module. Once the module prototype is successfully fabricated, it will be replicated to 2 modules. After the symmetry test, communication through RF Modules will be developed between the modules so that they are synchronized and are ready to be programmed. Genetic programming will be used to serve the purpose. Success in generating a program to execute a particular type of motion or formation will lead to the achieved of the project’s goal. Further improvements will be made according to the remaining time span.

Work Flow Chart



3.1) Hardware

The hardware design and fabrication along with electrical component selection process comprises of the following steps.

3.1.1) Mechanical Design

3.1.1.1) Initial Approach

Our reconfigurable robot, WICSER robot, is a chain based modular robot is one where all modules are connected into a long chain of modules. Before making the drawing on Auto-Cad different handmade prototypes are made.

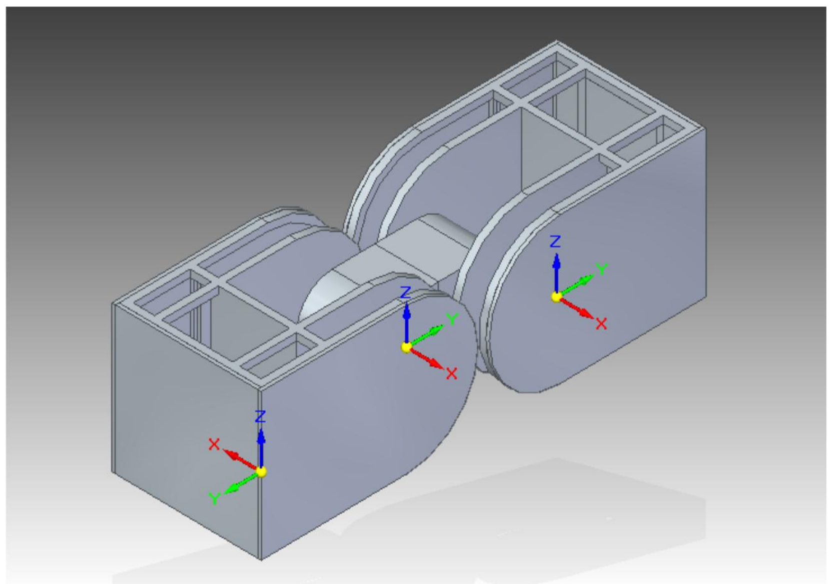


Figure 5 Solid Edge Design

All the configurations that are achieved by modular robot involve the movement of each module in a different fashion. As it is quoted, modules that are connected with each-other to develop morphologies must be designed to provide adequate motion and force.

3.1.1.2) Initial design and fabrication

Firstly, it is decided to manufacture the modules through rapid prototyping. So a rapid prototyping machine requires .stl file from Solid edge. The primary design of a block is showed in figure 1. But due to unavailability of rapid prototyping machine, it led to the fabrication of modules by acrylic using laser cutting.

3.1.2) Module Specifications

3.1.2.1) Degree of Freedom

In WICSER, modules are designed in such a manner that each module depicts two degree of freedom.

3.1.2.2) Symmetry of Modules

Like a single individual of a chain, each module has mechanical and electrical symmetry with other individuals. Each module is fabricated using 5mm thick acrylic sheet and for the sake of providing desired shape are molded upon heating.

3.1.2.3) Module Parts

The modules are connected using strong permanent Neodymium magnet. A module is comprised of two blocks i.e. passive and active. With the purpose of doing differentiation between passive and active blocks different colored acrylic was used. Formal one contains dark brown whereas lateral one has transparent acrylic for its body.

3.1.2.4) Modules Movement

Source of motion is precise position controlled servos. Two servo motors are embedded in an acrylic made link into up and down orientation to diminish the weight distribution factor. The motors provide semicircular motion to their respective blocks.

3.1.3) WICSER Robot Modules

The idea is to provide each module its own control unit, wireless communication source and power source. On its place each module is a self-complete robot but their robustness and versatility is aggravated when connected.

3.1.3.1) Control Unit

Control unit is Arduino UNO revision-3 that is programmed and interfaced with R.F. modules

3.1.3.2) Power Distribution

Power distribution comes with two batteries of different ratings with common ground. Control unit and R.F module are powered by 9 volts, on the other hand lithium 7.4 volts battery is dedicated for servos. While each servo requires 5 volts for their functioning, so a small Vero board PCB is designed to truncate the voltage to servos operating voltage levels.

3.1.3.3) Proximity Sensor

As per connection with other modules the permanent magnets two on each side are fixed. In order to perceive obstructions an I.R sensor is also incorporated, and in response to sensor signal modules change their outward features.

3.1.3.4) R.F. Modules

It is planned that through RF Wireless RF Transceiver Module 433Mhz CC1101 each module would be knowing where the next module is or how much angular displacement the blocks of any module has been moved. The wireless communication makes WICSER different from M-TRAN that has been made in history.

3.1.3.5) Material Selection

Selection of material for the body of modules requires some specification. The cost, availability, strength to weight ratio and feasibility with electrical components all are in favor of acrylic.

Acrylic Specification

The two acrylics sheets of 5mm thickness, dark brown and transparent are cut into two dimensional design.

3.1.3.6) AutoCAD design

The design of each block was made on AutoCad-2013 as shown in figure. The outer U-shaped periphery helps blocks to slide on one another and is of 2 inches diameter. There are two shaped peripheries, one is inside and other is outside. The outside periphery contains rectangular shaped cut outs to place magnets in them. The rectangular cut outs dimensions are 2cm x 1.2 cm. A quick observation tells that center of rectangles are not aligned. Each rectangle was designed under consideration that magnetic flux line may pass minimally through the servos as servo is having metal gearing inside. On external U-shape body magnets are fixed and there are small width and 1 inch length rectangles from where the body molded to 90 degree angle. There is another rectangle larger than previous described rectangles. It is planned to provide window for proximity sensor. In internal U design there are small circles. These circles are basically holes where motor's fan like mountings were fixed with the help of screws. There are numerous holes because many shaped motor mountings are present, so that motor can be fixed any way. If it is observed closely, the mid of internal U there are two holes on the upper side while one hole is on the right side down are present to fix Arduino UNO board. Between two these U's there is 1 inch long linkage to provide base to the block.

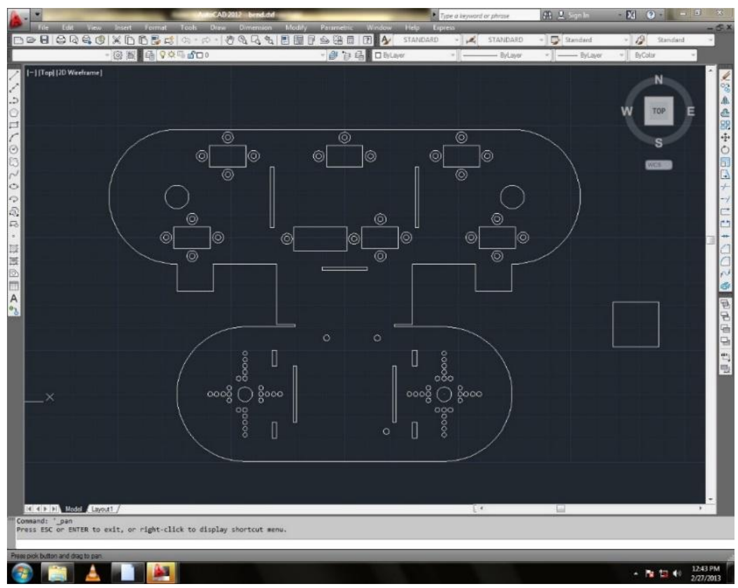


Figure 6 AutoCAD Design

3.1.3.7) Laser Fabrication

Once the design is completed on Auto-Cad, the .dxf file is exported and loaded into Corel draw software where the all boundaries are marked for laser operations. In Figure 3, same design that was created in AutoCAD environment is in Corel draw. The all boundaries are marked with different colors representing different laser operations.

Green Boundary

The green boundary commands the laser point for 5mm acrylic. This is the outer most boundary which has to be cut through the thickness.

Red Boundary

This boundary provides the instructions to laser operating system to cut fully through the Acrylic sheet lying inside the main boundary.

Blue Boundary

This boundary is committed to run the laser pen inside the main boundary but not making thorough cuts. Like engraving laser point varies its power to remove some portion of the acrylic. It was used to locate the head of screws inside so that when modules will connect, they will not provide any hindrance.

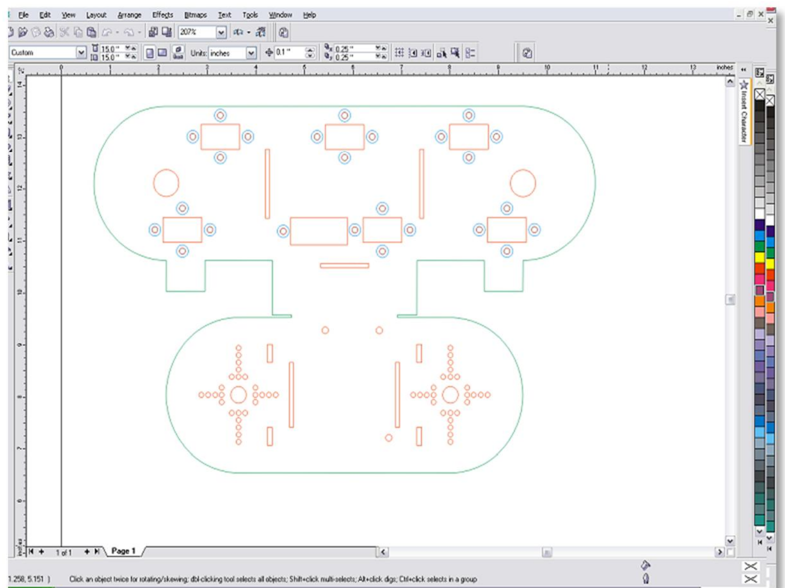


Figure 7 Corel Draw Design for Laser Fabrication

3.1.3.8) Bending Procedure

The result of laser cutting would be a two dimensional cutting sheet. So the third dimension, as shown in the Figure, in which the different bends are marked with alphabets. These bends are done by placing the side on the thin heater filament.

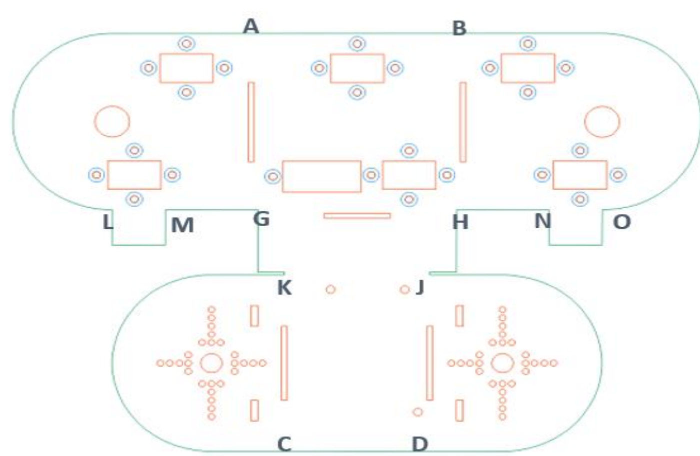


Figure 8 Marking of Points-to-Bend

Step I

Bend the AB and BH perpendicular

to AG and GH.

Step II

Bend KC and JD perpendicular to KJ and CD.

Step III

Bend GH as shown in figure 8.

Step IV

Now bend KJ in such a way that internal U comes inside the external U. The resulting output of bending is shown in figure 9.

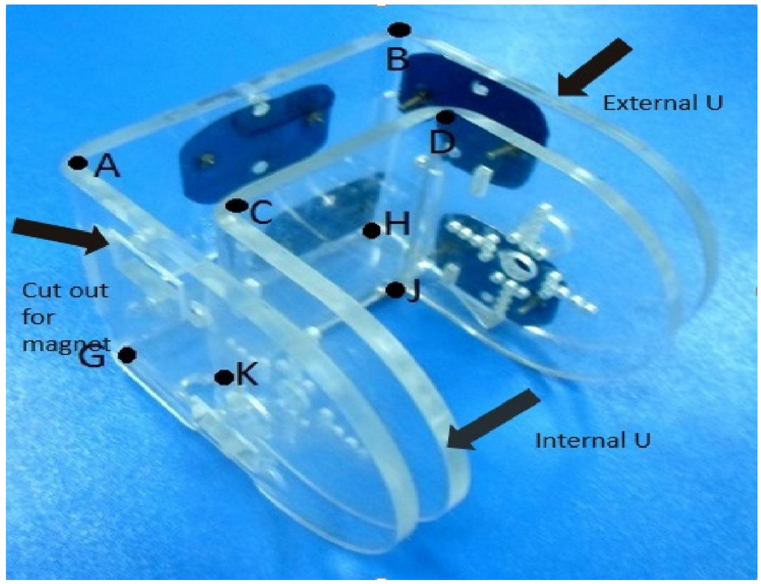


Figure 9 Bended Block

Caution

All module blocks are made by this way. Extreme care should be taken while bending in order to not lose symmetry.

3.1.3.9) Link Designing

Why Link?

Link is designed to hold motors. Two pockets are made in which motors are placed and these pockets are the screwed into a base.

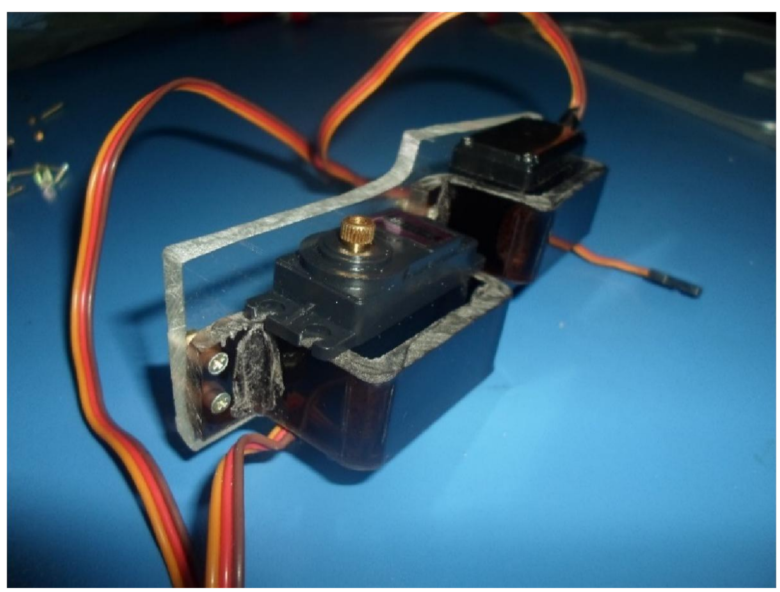


Figure 10 Link

Why motors are offset?

In order to cater weight and torque distribution factor motors in a single module are connected in opposite side of Internal U of active and passive block.

Link Design

Link design was simple and it was made my hands as shown in figure.

Motor's Alignment

Motor alignment with block is made by passing a screw from the pocket to hole of block.

3.1.3.10) Motor Linkage

Servo motor package contains circular or fan like mounting that can easily be attached with the object to be connected. These mountings have internal teeth that mesh with the external teeth of motor shaft.

The linkage has a quality that when motor is rotating its block if comes to stop than motor will not stop rotating. It will carry other block with its motor in the same module.

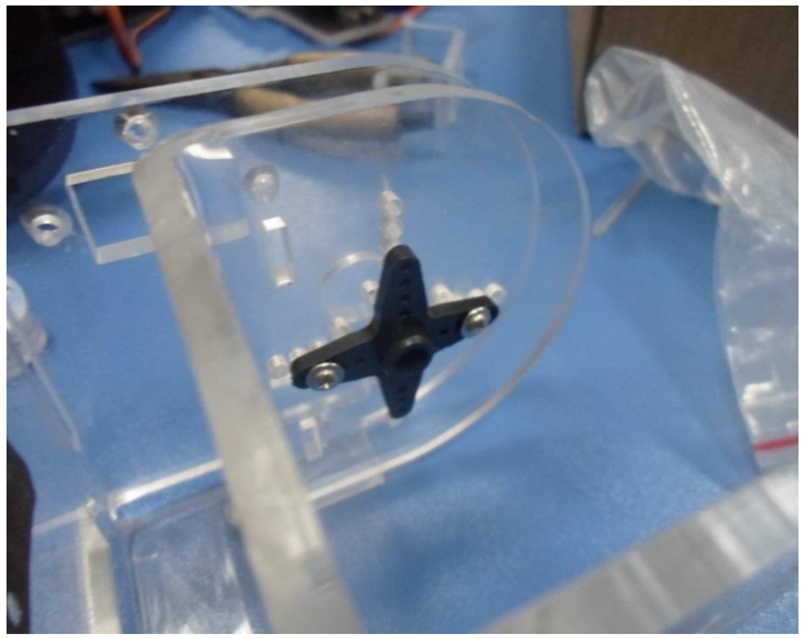


Figure 11 Motor Linkage

3.1.3.11) Magnetic shielding

First, one important point must be clear: Magnetic shielding does not block a magnetic field. No material can stop the lines of flux from traveling from a magnet's North Pole to its South Pole. The field can, however, be redirected.

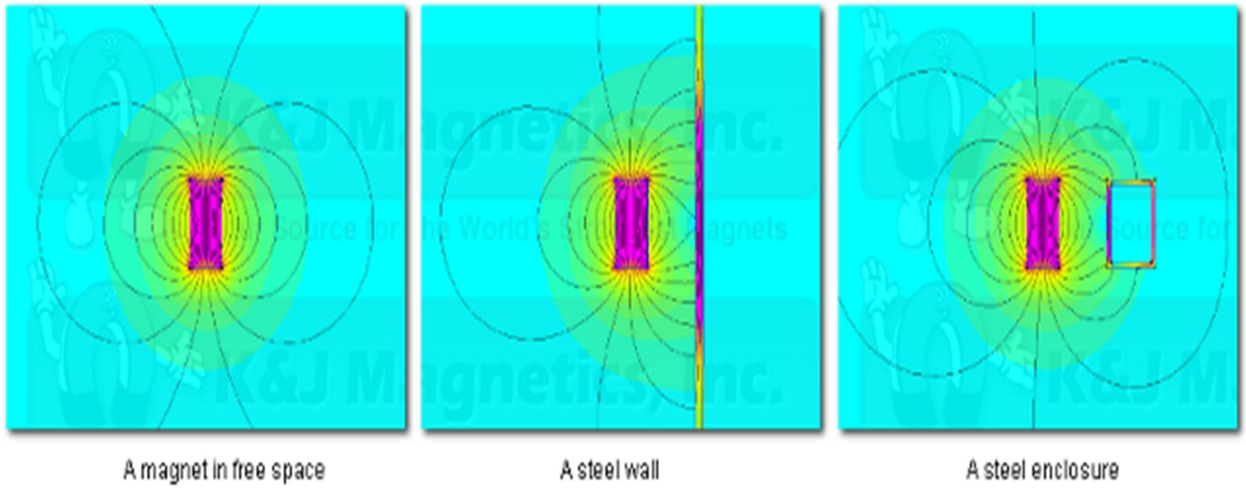


Figure 12 Shielding Effect by Steel Sheet

However a complete study and graphical observation of materials, steel provides best shielding. That's why small pieces are cut and fixed on the backside of side of magnets. Magnets are attached by using ordinary elfy glue. In present modules the magnets are attached on just one passive block and one active blocks of another module.

3.1.4) Complete Module

After assembling motors and magnets on the steel sheets, module is mechanically ready. There are different figures are given to show how it is looked like.

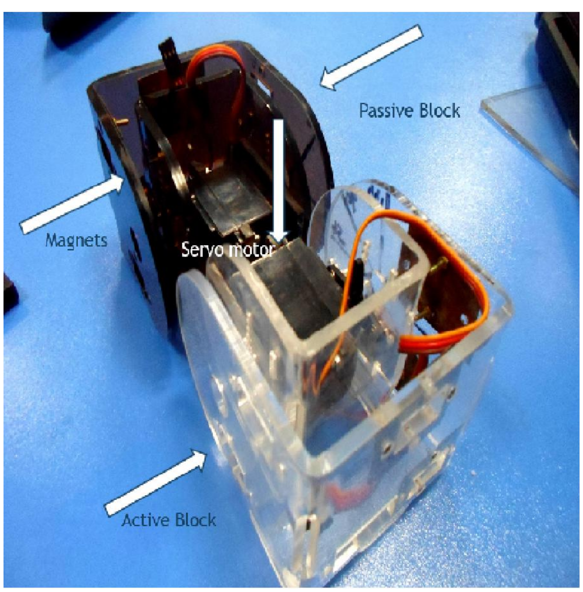


Figure 13 Complete Module-I

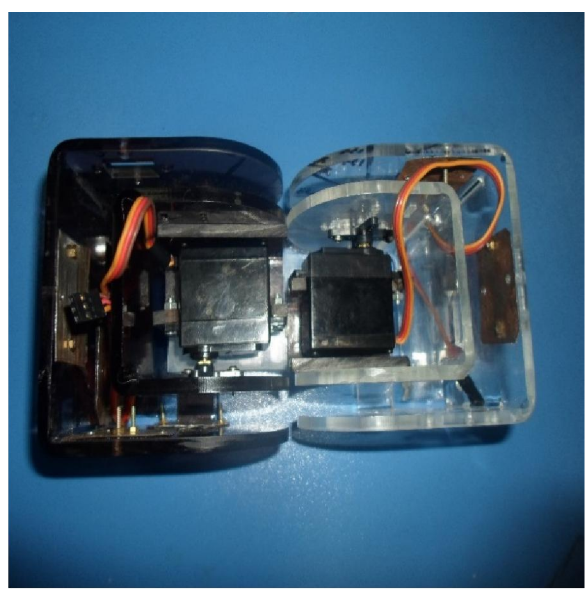


Figure 14 Complete Module-II

3.1.4.1) Single Module configuration

As it is discussed before that a module is a complete robot on its behalf. It can change its morphology, but changing one module configuration is vivid. However, when same modules are connected and when they transform it results into obvious and splendid transformation.

Single module can change its shape is shown in figure.

A single block is capable of lifting other block as shown in the figure

Figure 15 Single Module configuration-I

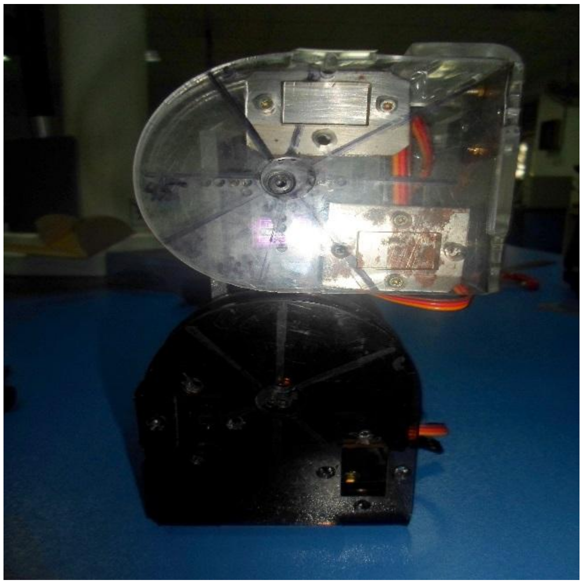


Figure 17 Single Module configuration-II

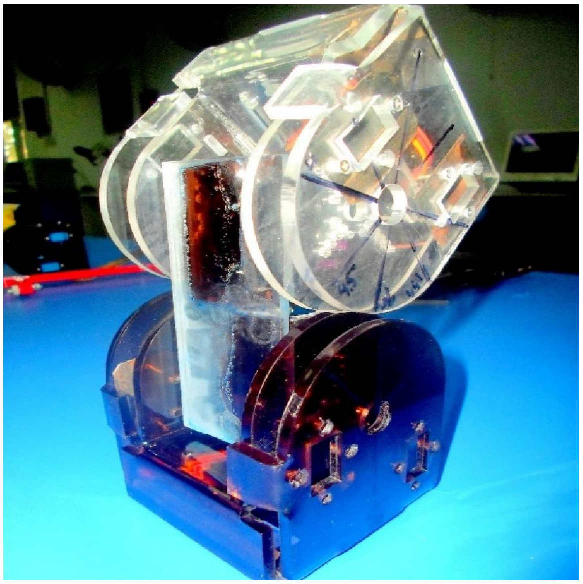


Figure 16 Single Module configuration-III

When the two modules are joined end to end they create morphology like a worm as shown in the figure.

Through this end to end connection our robot can take many shapes according to external circumstances.

Hence the mechanical part is completed. And the robot is ready for the building of its algorithm and electrical part.

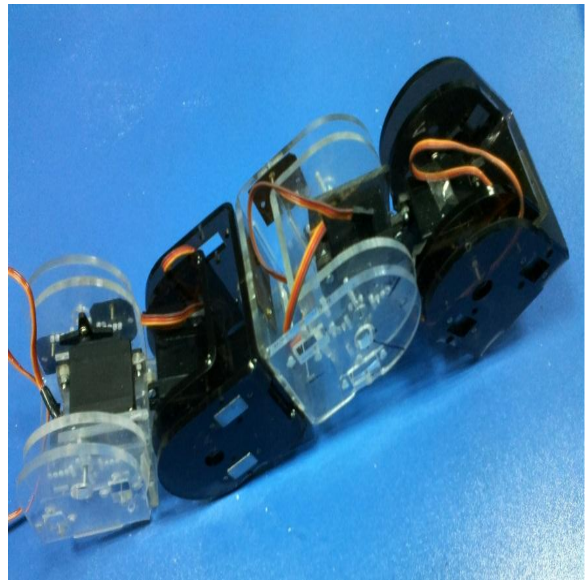


Figure 18 Joined Modules

Problems with Solutions in Final Design

- There must be some more space and support between internal and external U's.
- There should be some way so that magnets cut outs will be aligned so that any side can be attached to other side.
- Some internal supports should be there in so that less stress on bends.
- Bending must be done with extreme care otherwise holes will not be aligned and it will make assembly difficult.

3.1.5) Electrical Design

3.1.5.1) Power distribution

- There are two batteries are used one is Toshiba 9V battery and other is 7.4V lithium laptop battery.
- 9V battery provides power to control unit, Arduino Rev-3, R.F. modules and proximity I.R. sensor.
- Both batteries are common grounded so that Arduino signals for PWM can be provided to motors signal pins.
- 7.4V lithium laptop battery provides power to R.C. servos.
- R.C. servos operate on 5Volts so two PCBs are designed with common ground to truncate voltage levels from 7.4 volts to 5V

3.1.5.2) PCB design

- There are two PCBs in one module for both motors.
- The PCB design is shown in the figure with the components rating and voltages.
- Number of headers can be increased in order to facilitate more connections.
- PCB is made on Vero board.
- Fuse 0.8A is used to safe motors that can draw 1A current under full load.

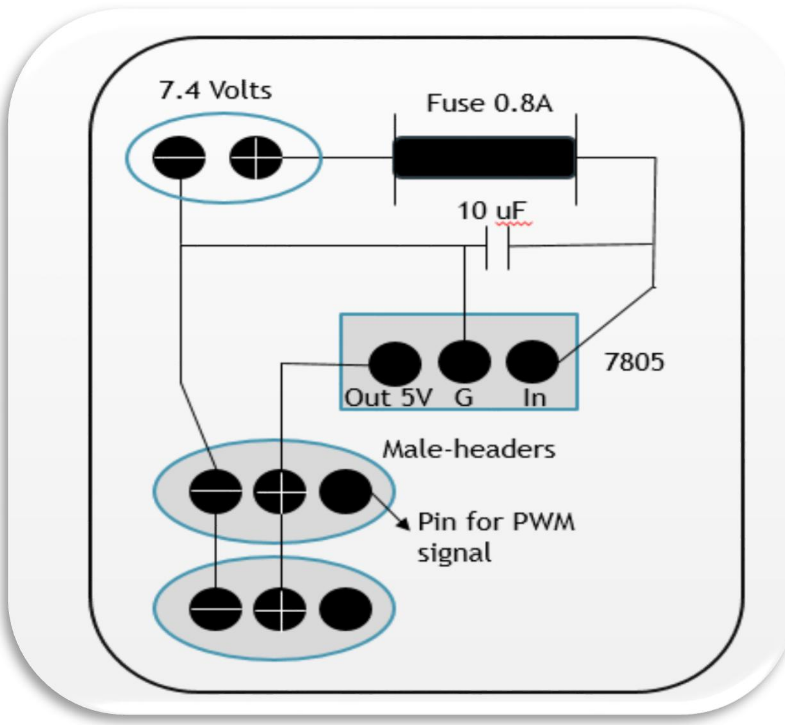


Figure 19 Power PCB Design

3.1.5.3) Arduino UNO Rev-3 board

Specifications

Microcontroller[23]	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 (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

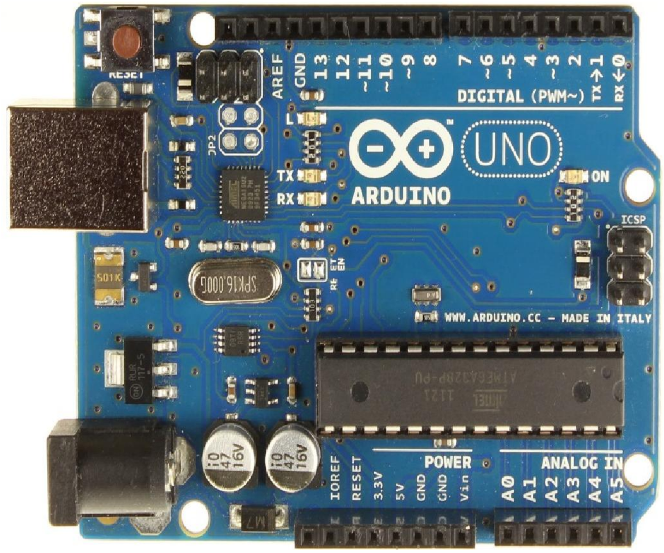


Figure 21 Arduino Board Front Side

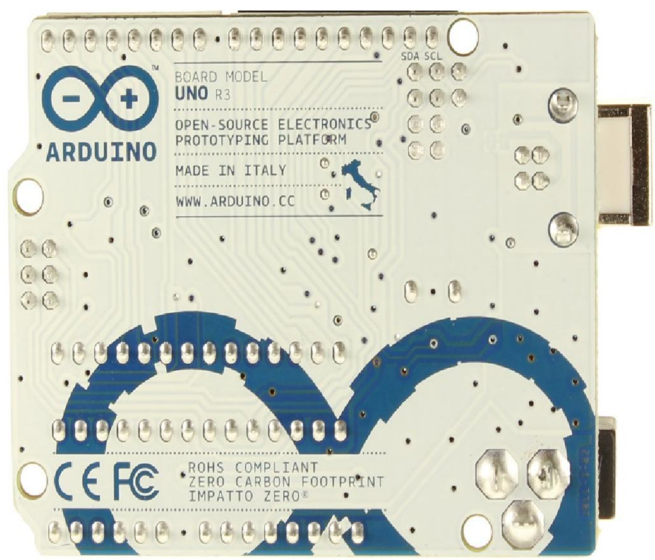


Figure 20 Arduino Board Rear Side

Details

Arduino UNO

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, 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 an AC-to-DC adapter or battery to get started.

"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.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and VIN pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

Power pin

The power pins are as follows:

VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins.

IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins.

USB Overall protection

The Arduino Uno has a resettable poly fuse that protects your computer's USB ports from shorts and over current. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

3.1.5.4) Pins Configuration in WICSER Robot

- Pin 10 is for Rx and pin 11 is for Tx connected to R.F. Module.
- Pin 8 is to provide PWM signal to white block motor.
- Pin 9 is to provide PWM signal to black block motor.
- RF gets its power from UNO.
- Arduino plus provides common ground to both batteries.

3.1.5.5) 433MHz RF Transceiver RF1100 CC1101 Wireless Module

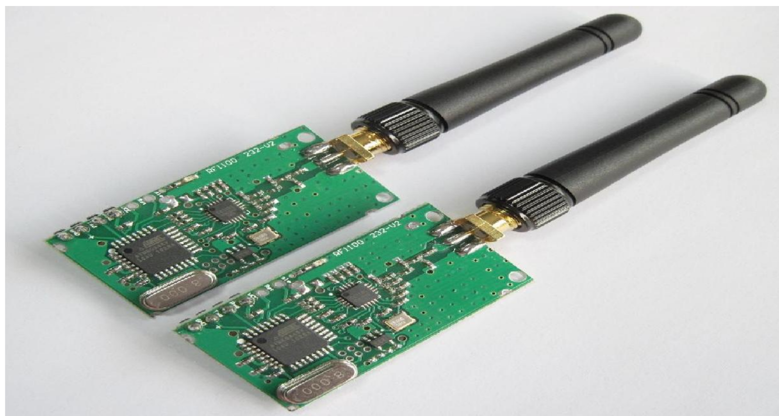


Figure 22 433MHz RF Transceiver RF1100 CC1101 Wireless Module

Introduction

- Micro-power wireless data transmission module.
- Using high-performance CC1101 wireless communication chip.
- The maximum transmission data rates up to 500Kbps, rate can modify by software.
- Coverage up to 300~500 meters in open areas.
- With a wireless wake-up features, sensitivity to -110dBm, high reliability,
- Working in the 433MHz frequency[24] band. Compliance ETSI（EN300-220-1 & EN301-439-3), Compliance the requirements of wireless control, no need to apply frequency usage license.

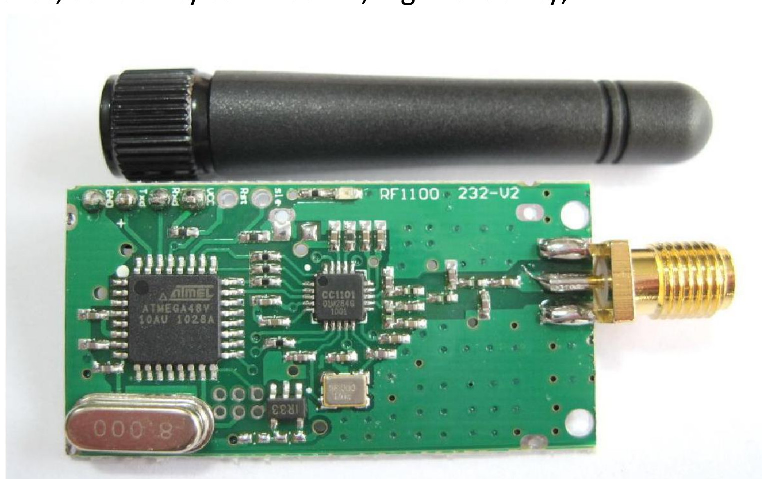


Figure 23 RF module with off antenna

Specification

- Operating voltage: 4.5~5.5V
- The maximum rate of 500kbps, support for 2-FSK, GFSK and MSK modulation
- Setup software can modify Rate, Power, Frequency
- High sensitivity (1.2Kbps under-110dBm, 1% packet error rate)
- Built-in hardware CRC error detection, and address control
- Lower current consumption (RX in, 15.6mA, 2.4Kbps, 433MHz)
- Programmable control of output power, up to +10 dB
- Wireless wake-up functionality to support low-power electromagnetic activation, wireless wake-up low-power sleep state of the equipment
- Support the transmission channel before the auto-clean-up visits (CCA), the Carrier Sense System
- Fast frequency changes brought about by a suitable synthesizer frequency hopping system
- Module can be software-based address, software programming is very convenient
- Standard 2.0mm DIP spacing interfaces for embedded applications
- Separate 64-byte RX and TX data FIFO
- Transmission Distance: open to the actual transmission distance 250-300 meters (depending on the specific situation of the environment and communication baud rate settings
- Dimensions: 58 mm (L) x 18 mm (W) x 12 mm (H)

Pin Description

Pin	Signal	Direction	Description
2	TxD	Output	Transmitted data
3	RxD	Input	Received data
4	Gnd	N/A	Signal ground
7	Vcc	Input	Power supply (optional)

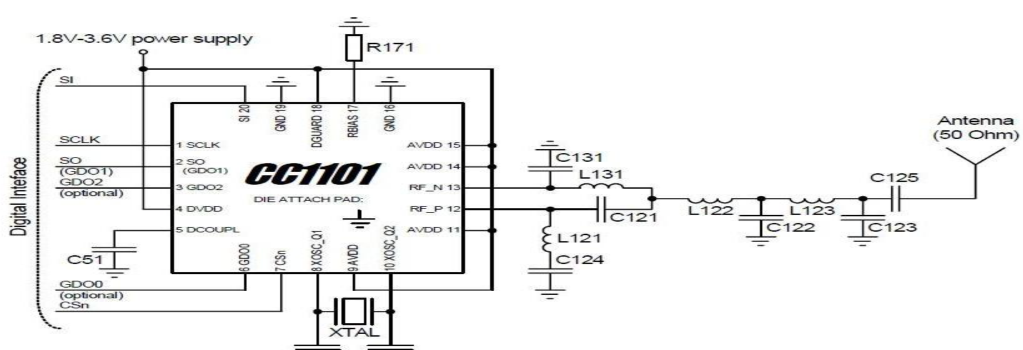


Figure 24 Circuit Diagram of CC1100

3.1.5.6) USB to RS232 TTL Converter module (PL2303)

- The USB to RS232 module provides the best and convenient way to connect your RS232 TTL Devices or demo board to your PC via the USB port.

Key Features

- Based on Prolific PL2303Hx chip[25]
- 5V Operation
- Driver support for Windows, Windows CE, MAC OS 8,9,X and Linux
- Compact Shell Cover
- 1 meter long cable
- 4 way JST connector (GND,TX,RX,DTR)
- Those who require the use of VCC/CTS/RTS/RI/DSI etc. may open the shell cover to locate required pads.

Figure 25 USB to TTL converter

Pin Out

- GND = Black Wire
- TX = Yellow Wire
- RX = Green Wire
- DTR = Blue Wire

Dimensions

- 1.7 * 5.5 cm

3.1.5.7) MG946 Compatible Metal Gear RC Servo

Description

- Compliant with most standard receiver[26] connector: Futaba, Hitec, Sanwa, GWS etc...
- Great for truck, Boat, Racing Car, Helicopter and Airplane.
- Stable and Shock Proof. - Coreless Motor - Metal Gear - Double Ball Bearing –
- Connector Wire Length 300mm
- Dimension : 40mm x 19mm x 43mm
- Weight : 55g
- Operating Speed : 0.17sec / 60 degrees (6.0V no load)
- Operating Speed : 0.20sec / 60 degrees (4.8V no load)
- Stall Torque : 10.5 kg-cm at 4.8V
- Stall Torque : 13 kg-cm at 6V
- Operation Voltage : 4.8 - 7.2Volts
- Gear Type: All Metal Gears
- Connector Wire: Heavy Duty

Figure 26 RC servo motor

3.1.5.8) Neodymium Magnets

Product Code[27]:	F201-5
Shape:	Rectangle
Length x Width	2cm x 1.2 cm
Thickness:	5mm
Plating:	Nickel
Material:	NdFeB
Vertical Pull (Kg):	3.8
Max Temp (degrees C):	80
Fixing:	Araldite/Loctite

Figure 27 Neodymium Magnets

3.1.5.9) Infrared Proximity Sensor Sharp GP2Y0A02YK0F

Description

Infrared proximity sensor made by Sharp. Part # GP2Y0A02YK0F has an analog output that varies from 2.8V at 15cm to 0.4V at 150cm[28] with a supply voltage between 4.5 and 5.5VDC. The sensor has a Japanese Solder-less Terminal (JST) Connector. This sensor is great for sensing objects up to 5 feet away.

Figure 28 IR sensor Pictorial view

Features

- Distance measuring range : 20 to 150 cm
- Analog output type
- Package size : 29.5×13×21.6 mm
- Consumption current : Typ. 33 mA
- Supply voltage : 4.5 to 5.5 V

Applications

- Touch-less switch
- Sensor for energy saving
- LCD monitor)
- Amusement equipment
- (Robot, Arcade game machine)

Characteristics

The characteristics of IR sharp sensor can be understood by following graph. By the graph it is obvious that there is inverse relation between distance and the output of IR sensor. For shorter distances voltage makes direct relation with distance, but for the detection of larger distances voltage goes down.

Henceforth, we are using large range sensors.

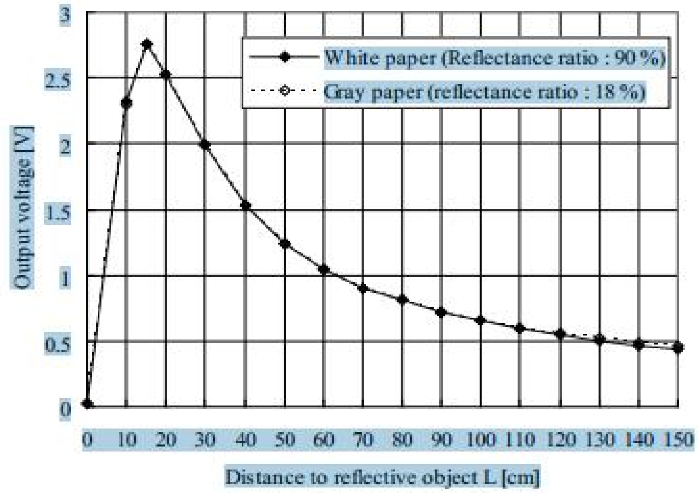


Figure 29 IR sensor Graphical Demonstration

Advice for the power supply

In order to stabilize power supply line, we recommend to insert a by-pass capacitor of 10µF or more between Vcc and GND near this product.

3.2) Software

3.2.1) MODULE 1

```
#include <SoftwareSerial.h>
#include <Servo.h>
Servo servowhite;       // create servo object to control a servo
Servo servoblack;       // create servo object to control a servo
int count = 1;
int pos = 0;
char check = 0;
SoftwareSerial mySerial(10, 11);
void setup()
{
  Serial.begin(9600);
  delay(1000);
  mySerial.begin(9600);
  delay(1000);
  pinMode(10, INPUT);
  servowhite.attach(8); // attaches the servo on pin 8 to the servo object
  servoblack.attach(9); // attaches the servo on pin 9 to the servo object
  while(!Serial)
  {
    ;
  }
}
void loop()
{
  if (Serial.available())
  {
    check= Serial.read();
    Serial.write(check);
  }
```

```
else
{
 check=check;
}
if (check=='p')
{
 Serial.write(check);
 mySerial.write('p');
 delay(100);
 pack();
}
else if (check=='w')
{
 Serial.write(check);
 mySerial.write('w');
 delay(100);
 worm();
}
else if (check=='u')
{
 Serial.write(check);
 mySerial.write('u');
 delay(100);
 unpack();
}
else if (check=='t')
{
 Serial.write(check);
 mySerial.write('t');
 delay(100);
 two_leg();
```

```
  }
}

void pack()
{
   servowhite.write(180);
   servoblack.write(180);
   delay(1000);
   mySerial.write('a');
   delay(100);
   while(!mySerial.available())
   {
     ;                //wait untill data is recieved by serial
   }
   if (mySerial.available())
   {
     char x = mySerial.read();
     if(x=='b')
     {
       servowhite.write(180);
       servoblack.write(180);
       delay(1000);
     }
   }
}

void unpack()
{
   servowhite.write(90);
   servoblack.write(90);
   delay(1000);
```

```
    mySerial.write('c');
    delay(100);
  }

  void worm()
  {
   while(count < 10)
   {
    mySerial.write('a');
    delay(1000);
     if (mySerial.available())
     {
      char x = mySerial.read();
      if (x=='a')
       {
        for(pos = 45; pos < 105; pos += 1)          // goes from 0 degrees to 180 degrees
        {                                           // in steps of 1 degree
          servowhite.write(pos);                    // tell servo to go to position in variable 'pos'
          delay(10);                                // waits 10ms for the servo to reach the position
        }
        for (pos = 100; pos > 10; pos -= 1)         // goes from 0 degrees to 180 degrees
        {                                           // in steps of 1 degree
          servoblack.write(pos);                    // tell servo to go to position in variable 'pos'
          delay(10);                                // waits 10ms for the servo to reach the position
        }
       }
       mySerial.write('a');
       delay(1000);
     }
     count+=1;
    }
```

```
  mySerial.write('z');
  delay(1000);
}

void two_leg()
{
  while(count==1)
  {
    servowhite.write(135);
    servoblack.write(90);
    delay(1000);
    mySerial.write('a');
    delay(100);
    count=count+1;
  }
    while(!mySerial.available())
    {
      ;              //wait untill data is recieved by serial comm
    }

    if (mySerial.available())
    {
      char x = mySerial.read();
      while(x!='z')
      {
        if(x=='b')
        {
          servowhite.write(90);
          servoblack.write(0);
          delay(1000);
          mySerial.write('c');
```

```
    delay(100);
   }
   if(x=='d')
   {
    servowhite.write(45);
    servoblack.write(45);
    delay(1000);
    mySerial.write('e');
    delay(100);
   }
   if (x=='f')
   {
    servowhite.write(0);
    servoblack.write(90);
    delay(1000);
    mySerial.write('g');
    delay(100);
   }
   if (mySerial.available())
   {
     x = mySerial.read();
   }
  }
  mySerial.write('z');
  delay(100);
 }
}
```

3.2.2) MODULE 2

```
#include <SoftwareSerial.h>
#include <Servo.h>
Servo servowhite;       // create servo object to control a servo
Servo servoblack;       // create servo object to control a servo
int pos =0;
int count = 1;
char check = 0;
SoftwareSerial mySerial(10, 11);
void setup()
{
  mySerial.begin(9600);
  delay(1000);
  pinMode(10, INPUT);
  servowhite.attach(8);   // attaches the servo on pin 8 to the servo object
  servoblack.attach(9);   // attaches the servo on pin 9 to the servo object
  while(!mySerial.available())
  {
    ;
  }
}
void loop()
{
  if (mySerial.available())
  {
    check= mySerial.read();
  }
  else
  {
    check=check;
  }
```

```
  if (check=='p')
  {
    pack();
  }
  else if (check=='w')
  {
    worm();
  }
  else if (check=='u')
  {
    unpack();
  }
  else if (check=='t')
  {
    two_leg();
  }
}
void pack()
{
   while(!mySerial.available())
   {
    ;                    //wait untill data is recieved by serial
   }
   if (mySerial.available())
    {
      char x = mySerial.read();
      if (x=='a')
      {
        servowhite.write(90);
        servoblack.write(20);
        delay(1000);
```

```
        servowhite.write(0);
        servoblack.write(20);
        delay(1000);
        mySerial.write('b');
        delay(100);
      }
    }
}
void unpack()
{
   while(!mySerial.available())
   {
    ;                    //wait untill data is recieved by serial
   }
   if (mySerial.available())
    {
      char x = mySerial.read();
      if (x=='c')
      {
        servowhite.write(90);
        servoblack.write(90);
        delay(1000);
      }
    }
}
void worm()
{
   if (mySerial.available())
   {
     char x = mySerial.read();
     while(x!='z')
```

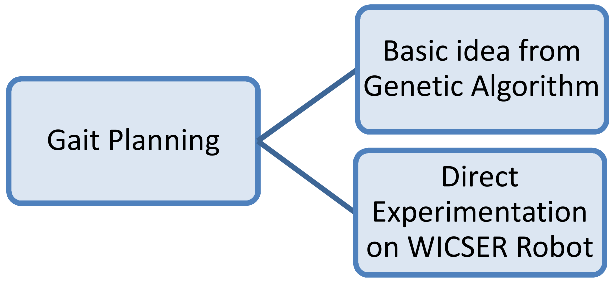
```
    {
     if (x=='a')
      {
       for(pos = 30; pos < 120; pos += 1)          // goes from 0 degrees to 180 degrees
       {                                           // in steps of 1 degree
         servoblack.write(pos);                   // tell servo to go to position in variable 'pos'
         delay(10);                               // waits 10ms for the servo to reach the position
       }
       for(pos = 10; pos < 120; pos += 1)         // goes from 0 degrees to 180 degrees
       {                                          // in steps of 1 degree
         servowhite.write(pos);                   // tell servo to go to position in variable 'pos'
         delay(10);                               // waits 10ms for the servo to reach the position
       }
       mySerial.write('a');
       delay(1000);
      }
      x=mySerial.read();
    }
   }
}

void two_leg()
{
  while(count==1)
  {
    servowhite.write(90);
    servoblack.write(90);
    count=count+1;
  }

   while(!mySerial.available())
```

```
{
 ;                    //wait untill data is recieved by serial
}
if (mySerial.available())
 {
   char x = mySerial.read();
   while(x!='z')
   {
     if (x=='a')
     {
       servowhite.write(90);
       servoblack.write(45);
       delay(1000);
       mySerial.write('b');
       delay(100);
     }
     if (x=='c')
     {
       servowhite.write(180);
       servoblack.write(90);
       delay(1000);
       mySerial.write('d');
       delay(100);
     }
     if (x=='e')
     {
       servowhite.write(135);
       servoblack.write(135);
       delay(1000);
       mySerial.write('f');
       delay(100);
```

```
        }
        if (x=='g')
        {
          servowhite.write(90);
          servoblack.write(180);
          delay(1000);
          mySerial.write('z');
          delay(100);
        }
        if (mySerial.available())
        {
           x = mySerial.read();
        }
    }
  }
}
```

Chapter 4: Gait Planning

4.1) Introduction



Following Gaits can be executed with the WICSER Robot.

I. Packing and Un-packing of modules
II. Worm Like Motion
III. Two-legged Walker

4.1.1) Packing and Un-packing of modules

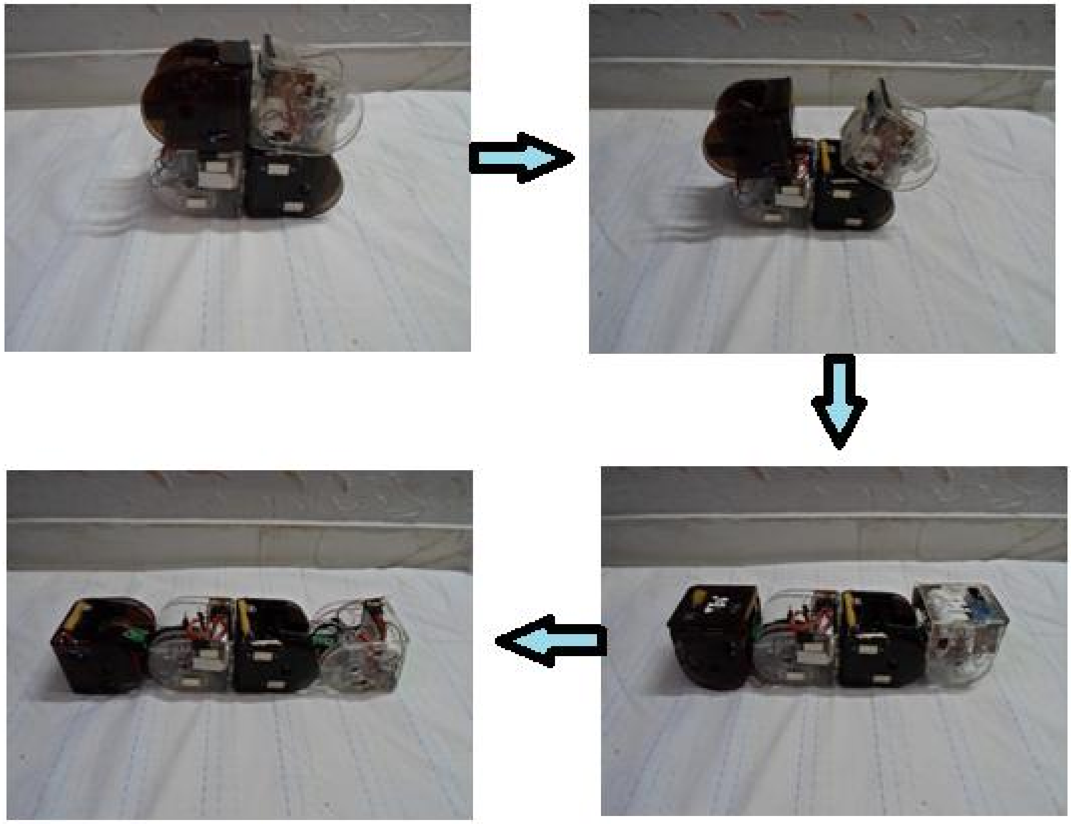


Figure 30 Pack to worm-type Configuration-I

4.1.2) Worm-Like Motion

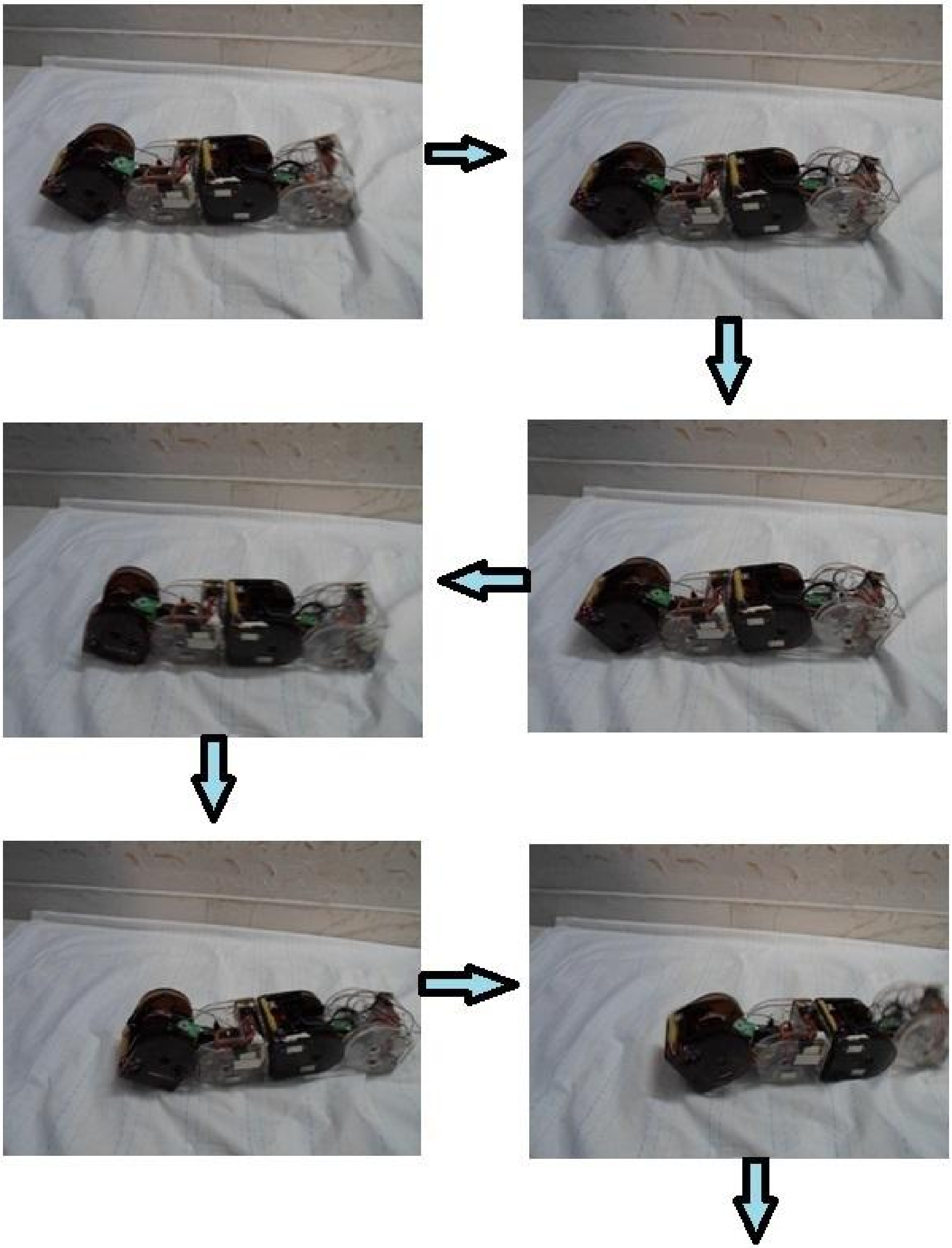


Figure 31 worm-type Reconfiguration

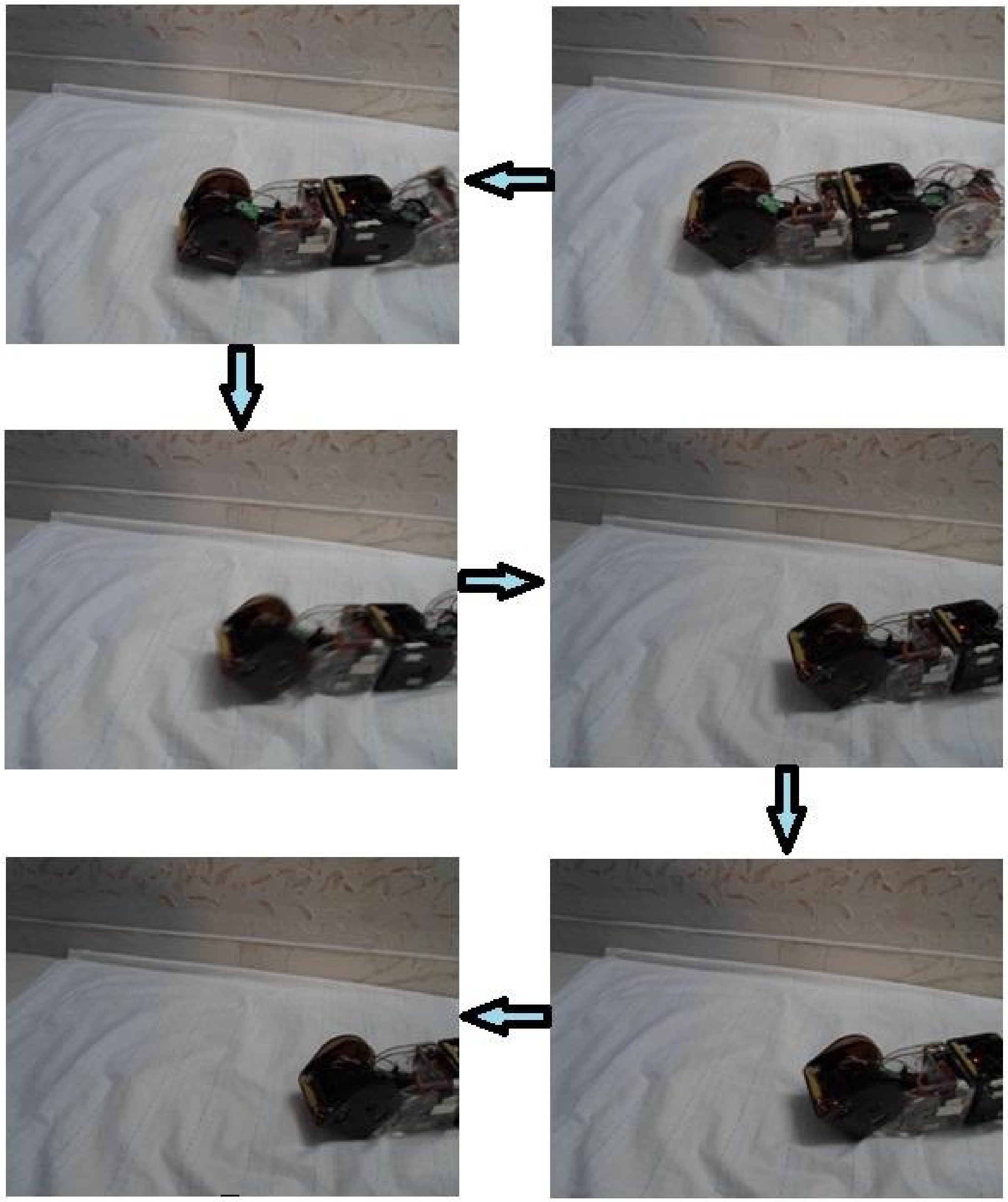


Figure 32 Worm-type motion Configuration

4.1.3) Two-Legged Walker

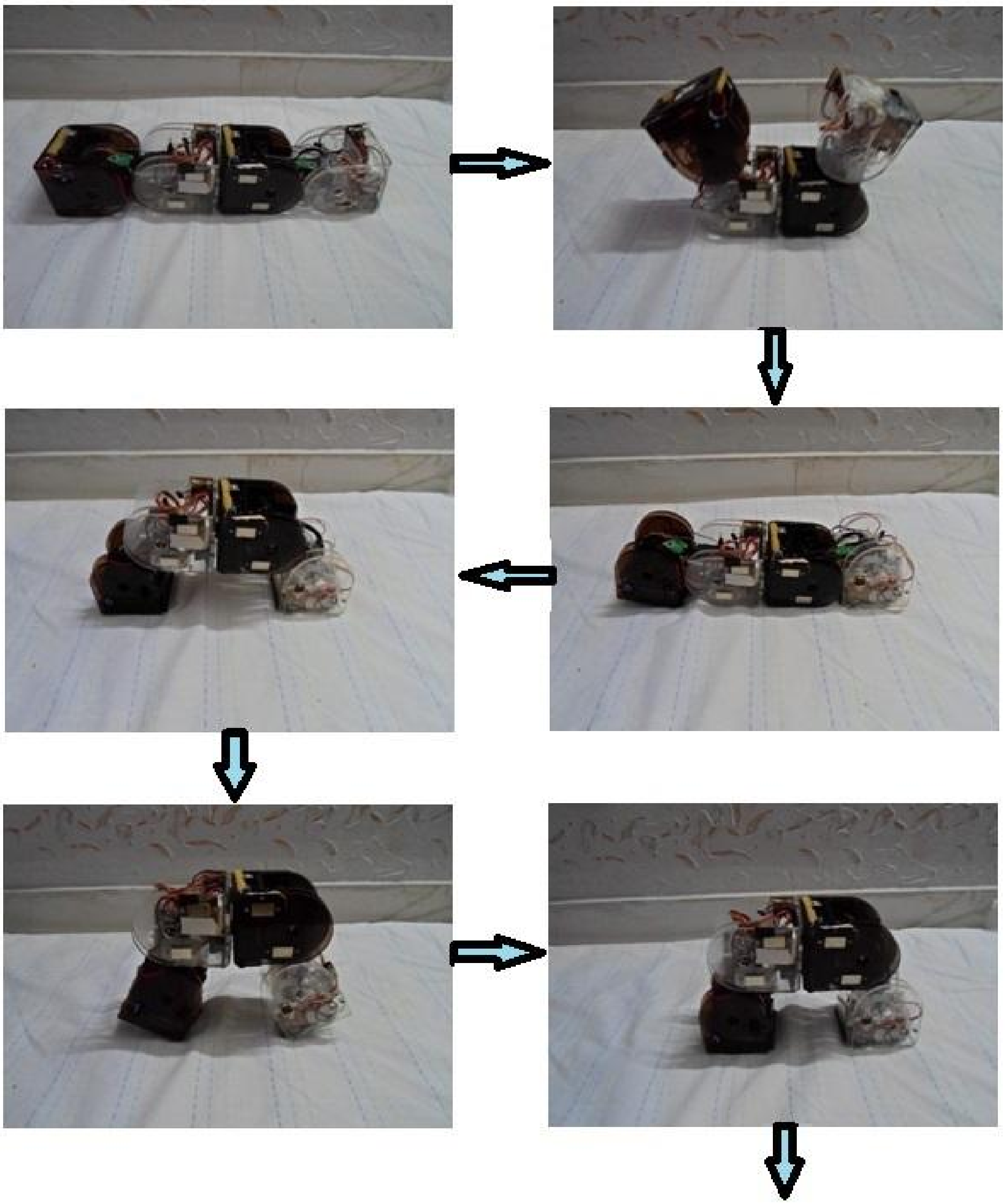


Figure 33 Worm to Walker Reconfiguration

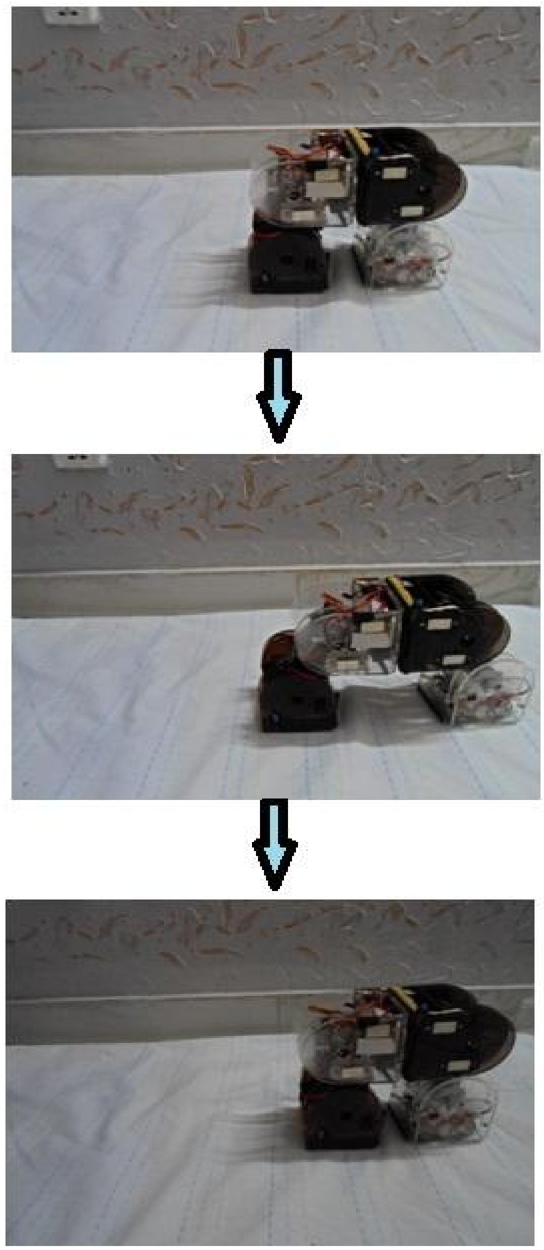


Figure 34 Walker Moving

Chapter 5: Conclusion and improvements

5.1) Conclusions

Completed tasks are as follows.

The Design and Fabrication of module was completed initially. Electrical and Mechanical Circuitry has been installed. Then the wireless communication through RF-module has been established and following configurations has been successfully executed.

- Packing and Un-packing configuration execution
- Worm-Like motion Configuration
- Two-Legged Walker configuration

The only Incomplete task in our project is the 'Automatic module detection through IR sensor' which we couldn't do because of the lack of time.

5.2) Improvements

The following improvements can be made to this project to make it more effective, appealing and efficient.

More modules can be fabricated and to execute many other different configuration
Gait algorithms can be made more efficient and intelligent.
Size can be changed to suit any particular task according to requirement.
It can also be made 3 Dimensional by adding another motor.
More magnets can be added in the remaining slots to execute other configurations.
By adding a gripper to one of the module, it can be used in different machine processes.
Automatic detachment mechanism can be installed on the modules through SMA coils.

References

[1] Murata, Satoshi, and Haruhisa Kurokawa. "Self-reconfigurable robots."*Robotics & Automation Magazine, IEEE* 14.1 (2007): 71-78.

[2] Salemi, B., M. Moll, et al. (2006). SUPERBOT: A Deployable, Multi-Functional, and Modular Self-Reconfigurable Robotic System. Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on.

[3] CapcarrÃ¨re, M., A. Freitas, et al. (2005). Co-evolution of Structures and Controllers for Neubot Underwater Modular Robots. Advances in Artificial Life, Springer Berlin Heidelberg. **3630:** 189-199.

[4] Yoshida, E., S. Murata, et al. (2003). Evolutionary synthesis of dynamic motion and reconfiguration process for a modular robot M-TRAN. Computational Intelligence in Robotics and Automation, 2003. Proceedings. 2003 IEEE International Symposium on.

[5] Butler, Z., K. Kotay, et al. (2004). "Generic Decentralized Control for Lattice-Based Self-Reconfigurable Robots." The International Journal of Robotics Research **23**(9): 919-937.

[6] Ricardo Franco Mendoza, G., S. Kasper, et al. (2007). A self-reconfigurable communication network for modular robots. Proceedings of the 1st international conference on Robot communication and coordination. Athens, Greece, IEEE Press.

[7] Yim, M., D. G. Duff, et al. (2000). PolyBot: a modular reconfigurable robot. Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on.

[8] Shen, W.-M., M. Krivokon, et al. (2006). "Multimode locomotion via SuperBot reconfigurable robots." Autonomous Robots **20**(2): 165-177.

---

[9] Butler, Z., K. Kotay, et al. (2002). Generic decentralized control for a class of self-reconfigurable robots. Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on.

[10] Kamimura, A., H. Kurokawa, et al. (2005). "Automatic locomotion design and experiments for a Modular robotic system." Mechatronics, IEEE/ASME Transactions on **10**(3): 314-325.

[11] Kamimura, A., H. Kurokawa, et al. (2003). Automatic locomotion pattern generation for modular robots. Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on.

[12] Kurokawa, H., A. Kamimura, et al. (2002). Self-reconfigurable modular robot (M-TRAN) and its motion design. Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on.

[13] Murata, S., E. Yoshida, et al. (2002). "M-TRAN: self-reconfigurable modular robotic system." Mechatronics, IEEE/ASME Transactions on **7**(4): 431-441.

[14] Yoshida, E., S. Murata, et al. (2003). Self-reconfigurable modular robots -hardware and software development in AIST. Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on.

[15] Wei-Min, S., B. Salemi, et al. (2002). "Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots." Robotics and Automation, IEEE Transactions on **18**(5): 700-712.

[16] Rico Moeckel, Cyril Jaquier, Kevin Drapel, Elmar Dittrich, Andres Upegui, Auke Jan Ijspeert, (2006) "Exploring adaptive locomotion with YaMoR, a novel autonomous modular robot with Bluetooth interface", Industrial Robot: An International Journal, Vol. 33 Iss: 4, pp.285 - 290

[17] Murata, S., E. Yoshida, et al. (2002). "M-TRAN: self-reconfigurable modular robotic system." Mechatronics, IEEE/ASME Transactions on **7**(4): 431-441.

[18] Yoshida, E., S. Murata, et al. (2003). Self-reconfigurable modular robots -hardware and software development in AIST. Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on.

[19] Kurokawa, H., A. Kamimura, et al. (2002). Self-reconfigurable modular robot (M-TRAN) and its motion design. Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on.

[20] Modular Self-Reconfigurable Robot Systems: Challenges and Opportunities for the Future, by Yim, Shen, Salemi, Rus, Moll, Lipson, Klavins & Chirikjian, published in IEEE Robotics & Automation Magazine March 2007

[21] http://en.wikipedia.org/wiki/Self-reconfiguring_modular_robot#cite_note-5

[22] Yim, M., D. G. Duff, et al. (2000). PolyBot: a modular reconfigurable robot. Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on.

[23] http://arduino.cc/en/Main/arduinoBoardUno

[24] http://www.goodluckbuy.com/433mhz-wireless-rf-transceiver-module-cc1101-cc1100-w-antenna.html

[25] http://digipak.org/zencart/index.php?

[26] http://pakelectronics.com.pk/index.php?route=product/product&product_id=251

[27]http://www.supermagnetman.net/product_info.php?cPath=39&products_id=1186&osCsid=6795295fc61f964cebfcfe89500000e8

[28] https://www.sparkfun.com/products/8958