HETEROGENEOUS ROBOT SWARM – HARDWARE DESIGN AND IMPLEMENTATION

Madhav Devidas Patil

Under the Supervision of Dr. Tarek Sobh

DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIRMENTS

FOR THE DEGREE OF DOCTOR OF PHILOSOHPY IN COMPUTER SCIENCE

AND ENGINEERING

THE SCHOOL OF ENGINEERING

UNIVERSITY OF BRIDGEPORT

CONNECTICUT

December, 2014

HETEROGENEOUS ROBOT SWARM – HARDWARE

DESIGN AND IMPLEMENTATION

Approvals

Committee Members

Name	Signature	Date
Dr. Tarek M. Sobh		9/3/15
Dr. Xingguo Xiong		08/25/2015
Dr. Hassan Bajwa		08/25/2015
Dr. Zheng Li		08/25/2015
Prof. Jack Toporovsky		08/27/2015
Dr. Devdas Shetty		August 13, 2015

Ph.D. Program Coordinator

Dr. Khaled M. Elleithy		9/2/15

Chairman, Computer Science and Engineering Department

Dr. Ausif Mahmood		8-31-2015

Dean, School of Engineering

Dr. Tarek M. Sobh		9/3/2015

HETEROGENEOUS ROBOT SWARM – HARDWARE DESIGN AND IMPLEMENTATION

HETEROGENEOUS ROBOT SWARM – HARDWARE DESIGN AND IMPLEMENTATION

ABSTRACT

Swarm robotics is one the most fascinating, new research areas in the field of robotics, and one of it's grand challenge is the design of swarm robots that are both heterogeneous and self-sufficient. This can be crucial for robots exposed to environments that are unstructured or not easily accessible for a human operator, such as a collapsed building, the deep sea, or the surface of another planet. In Swarm robotics; self-assembly, self-reconfigurability and self-replication are among the most important characteristics as they can add extra capabilities and functionality to the robots besides the robustness, flexibility and scalability. Developing a swarm robot system with heterogeneity and larger behavioral repertoire is addressed in this work.

This project is a comprehensive study of the hardware architecture of the homogeneous robot swarm and several problems related to the important aspects of robot's hardware, such as: sensory units, communication among the modules, and hardware components. Most of the hardware platforms used in the swarm robot system are homogeneous and use centralized control architecture for task completion. The hardware architecture is designed and implemented for UB heterogeneous robot swarm with both decentralized and centralized control, depending on the task requirement. Each

robot in the UB heterogeneous swarm is equipped with different sensors, actuators, microcontroller and communication modules, which makes them distinct from each other from a hardware point of view. The methodology provides detailed guidelines in designing and implementing the hardware architecture of the heterogeneous UB robot swarm with plug and play approach. We divided the design module into three main categories - sensory modules, locomotion and manipulation, communication and control.

We conjecture that the hardware architecture of heterogeneous swarm robots implemented in this work is the most sophisticated and modular design to date.

ACKNOWLEDGEMENTS

My thanks wholly devoted to God who has helped me all the way to complete this work successfully. I cannot express in words my gratitude to God for his countless blessings that made me able to pursue this highest degree in engineering.

I am really grateful to my dad and mom (Devidas, and Surekha) for their continuous support, understanding and encouragement during this whole journey.

Time flies! My first day in the United States and days spent at University of Bridgeport, are still fresh in my memory. Today, in finishing my doctoral studies, I see many people around me whose continuous support I will always regard, and therefore I would like to use this opportunity to express my gratitude to them.

First, I would like to thank my advisor Prof. Dr. Tarek Sobh for giving me the opportunity to pursue doctoral studies at RISC Lab, School of Engineering, University of Bridgeport. His valuable guidance and criticism during this time always helped me to develop new concepts and foster my research aptitude. He gave me ample freedom to explore and develop new concepts in the area.

I am really grateful to Prof. Jack Toporovsky for his valuable guidance during the time. I will also not forget the time with my friend Dr. Sarosh Patel, especially during writing this dissertation.

I am also thankful to my friend and fellow PhD candidate Tamer Abukhalil for the effort he has put in developing software platform for robot agents. I would not have completed my research without the software platform that he designed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Over the past decade, an increasing number of research and development activities related to modular swarm robotics are attracting considerable attention and interest in both academia and industry. This interest inspired by, among other things, the emergent behavior observed in social insects such as ants, bees, wasps, termites, etc. [**1**]. Self-reconfiguration, Self-assembly and Self-replication are the main distinguishing characteristics of swarm robots, and a dream long held by many researchers in the field of robotics is to develop fully autonomous robotic systems with these characteristics [**2**]. As with many new technologies, this field is growing rapidly and becoming more complex, but there remains much to accomplish in the development of swarm robotics hardware, as the performance of a swarm robotic system depends greatly on its mechanical and electronic control design [**3**]. With increasing system complexity, each robot must still follow simple rules to perform a task or any application.

The hardware design of a robot swarm is difficult to achieve, because designer needs to define the hardware and behavior for individual robot in the swarm. Evolutionary swarm robotics represents an effective way of designing robot swarm systems, however, those evolutionary techniques has been applied almost exclusively to homogeneous robot swarm systems. This work focuses on the design and implementation of a heterogeneous robot swarm.

1.1 Problem Statement

The performance of any machine or interoperable group of machines is highly dependent on its hardware architecture, which in turn depends on the overall mechanical and electronic control design and structure. The Swarm systems inherit all the challenges in designing the hardware architecture for individual robots such that each robot can perform a desired task in an unstructured dynamic environment. Moreover, several issues arise in order to coordinate the behavior for a swarm of robots to complete a task. For example, are the controls of group architectures of multi-robot systems centralized or decentralized? Is each robot in the group identical (homogeneous) or different (heterogeneous)? How can the robots resolve the resource conflict problem in a shared environment? What type of communication technique is suitable for specific tasks, do the robots need to exchange information explicitly or implicitly? To what degree should robots cooperate in order to accomplish the task? A general question in designing the hardware architecture of robot swarms is whether a specialized hardware needed for each task or whether more general hardware architecture can be developed which can be used for a wide range of applications. Is the heterogeneous hardware architecture of robot swarms effective enough for localization, mapping, exploring, and rescue?

Using heterogeneous swarm robots, however, makes the design procedure more challenging. Over the past several decades, numerous hardware architectures have been designed and developed for self-reconfigurable swarm robots. Each structure has focused on a different set of factors such as: flexibility, degrees of freedom, torque to weight ratio, power consumption, cost, size, control mechanism, etc. However, there are some fundamental, inherent limitations imposed by various architectures that can have a profound effect on how control and manipulation of autonomous mobile swarms is accomplished. Can we use heterogeneous swarm robot in real time constraints to work under the unknown environment by increasing the complexity of task? These architectural limitations can affect the precision of robot movement, robot strength, and the ruggedness of docking interfaces between modules. Motor power, power management, and the speed with which individual robots can move are also limiting factors on the performance of the reconfigurable swarm robot system.

1.2 Motivation

Since the days of early research in swarm robotics, the field has grown rapidly with much wider topics being analyzed and addressed. Prior to this most of the research concentrated on software design and algorithm implementation, with few of the hardware platforms developed for the robot swarm systems. Certain tasks may be too complex to be accomplished by a single robot no matter how capable the robot is. A single robot is able to complete a task in a designated amount of time easily, the challenge comes when coordinating multiple robots to complete the task together in order to improve the efficiency of the system. Building and using several simple robots may be easier, cheaper, more flexible and more fault-tolerant than having a single powerful robot.

Most of the work in swarm robots is based on homogeneous swarm architecture. These homogeneous robots are identical in size, shape, design and built using the similar hardware components. There are other issues related to the hardware design of the swarm robots are size, cost, weight, flexibility and task efficiency. To solve these open issues in the swarm robotics is a great challenge for the researchers in both hardware as well as software architecture.

1.3 Research Contribution

Following are the significant contributions of this work:

1. This work shows the design and implementation of heterogeneous robot swarm, consisting of different sensory units, actuation and communication units on each robot.

2. The UB Swarm of heterogeneous robots devises a hybrid distributed system that overcomes the drawbacks of both centralized and decentralized schemes.

3. This heterogeneous swarm system can carry out a large number of tasks simultaneously with simpler and cheaper robots, than a single sophisticated robot.

4. The power management system with fault tolerant is proposed and developed.

5. This hardware architectural design provides a non-expert user with an accessible yet very robust robotic platform in which it is easy to further add actuators and sensing modules without having to redesign.

6. Till date, to the best of our knowledge, there are only few heterogeneous swarm systems, but they are very expensive, complicated and require an expert to operate them. The proposed swarm robots are inexpensive, user friendly and can be used for any task.

7. These heterogeneous swarm robots which can be used for research purposes as well as for real life applications.

1.4 Organization of the Thesis

Chapter two is a literature survey of existing hardware platforms for swarm robots. This chapter presents the different hardware with sensory module, locomotion module, communication module, and power supply that has been used to design and implement the swarm systems.

Chapter three discusses the research plan of the UB Swarm system. In this chapter, the technical specification and working of all the components used for UB Swarm system is given in detail.

Chapter four discusses the hardware design and implementation of the UB Swarm system. The architectural design and implementation of hardware platform with power consumption and power management techniques are also presented.

Chapter five presents the result of the UB Swarm system. The experimental task given to UB Swarm robots such as obstacle avoidance, mapping, and human rescue is also discussed in this chapter.

Chapter six summarizes the work and draws some conclusions.

Finally, in chapter seven future works are presented.

CHAPTER 2: LITERATURE SURVEY

2.1 Introduction

In this section, we provide a brief survey of related work on swarm robot: self-reconfiguration, self-replication and self-reassembly. Modular robots are still in the process of becoming more flexible, autonomous, and more robust [**4**], [**5**]. Like any other robot, a swarm robot has two main organs; hardware and software. Software is the brain of the system, which gives a simulation environment to the functioning of the robot. The hardware brings directions stimulated by the software into action. When many such inter-communicating robots are deployed to work together, swarming action comes into play. However, only limited hardware platforms have been developed and used so far.

Swarm robots are usually homogeneous and controlled by a centralized or hierarchical system, depending on the application. Most of the robot platforms used in such swarm have the capability to assemble themselves according to the requirements of the task. Self-assembly is a process in which a group of swarm robots comes together to form a temporary large body structure capable of performing a job that is beyond the capability of single robot [**6**]. Christensen, O'Grady, and Dorigo describe a robotic system that exhibits this kind of self-assembly. In this system, the basic units are themselves robots that can function either independently when disconnected from one another, or they can function collectively when connected together to form a metastructure. The Christensen/O'Grady/Dorigo system demonstrates this kind of transformation of a collection of independent robots through a variety of different metastructure morphologies in physical hardware. Given enough units, if any individual unit in such a metastructure fails, the system would self-repair by replacing nonfunctional units with functional ones.

Almost seventy years ago, in 1947, Von Neumann proposed an automaton model sufficiently complex to reproduce itself [**7**], [**8**], [**9**]. Self-replication is another one of the characteristics of a modern swarm robot, in which several robot modules connect with each other to form an exact copy of the original robot [**4**]. The concept of kinematic self-reproduction has been applied in many research areas such as cellular automata, nanotechnology, macromolecular chemistry, and computer simulations. In the 1950s and 1960s, Penrose presented the first implementation of a passive self-replicating machine. He showed that simple units or "bricks" having certain properties could be employed to build a self-reproducing machine under external agitation. The replicated robot function is the same as the original robot so that it can perform the same task. Only a few, high-level modules have successfully demonstrated the ability to self-replicate, primarily due to the great complexity of the process. Such a process is extremely challenging for low level modular robots.

Self-reconfiguration is a process by which a robot metastructure constructed from physical structures or subsystems of modular robots, autonomously self-organizes and changes shape in order to adapt to different tasks or classes of terrain[**10**]. For instance, some modular robots may transform into snakes in order to follow a tunnel and then may transform into quadrupeds to go up stairs. In self reconfiguring, swarm modules are able to connect and disconnect without any human interaction as they offers such advantages as versatility, adaptability, robustness and cheap production over traditional robots [**11**], [**12**]. Due to these advantages, swarm robots exhibiting self-reconfigurability and self-assembly can be used to handle a wide range of tasks in an unknown or dynamic environment such as search and rescue operations after a fire or earthquake, undersea mining, planetary exploration, battlefield reconnaissance, and other application like service robotics and entertainment. Self-reconfiguration of a homogeneous system is simpler than in a heterogeneous system, but a heterogeneous swarm robot system might be more time-efficient at accomplishing certain tasks. Because the modular robot metastructure created by such a swarm system will be more compact due to the specialized capabilities of the modules [**13**].

According to [**5**], self-reconfigurable robots are classified into three main types: chain, lattice and mobile reconfiguration systems. In the chain and lattice types, each module typically remains connected to the (larger) modular robot at one or more points, while in mobile, modular systems, the system self-reconfigures by having modules detach themselves from the modular robot and move independently to another location to reconnect. Self-reconfigurable robots have proven to be capable of self-repair [**14**], [**15**], self-assembly, and locomotion over a either a plane surface or over widely varied terrain [**16**].

2.2 Hardware Platform Classification

To date, many sophisticated swarm robot platforms have been built by considering cost and functionality along with flexible distributed intelligence methods. Some examples are:

2.2.1 Lattice-based robot architecture

In lattice architectures, the mobile robot units are connected and arranged in regular three dimensional cubic or hexagonal grid patterns. The lattice architecture offers relatively simpler reconfiguration and control, since motion is accomplished in parallel within an open loop framework. Homogeneous “molecubes” based on a lattice self-reconfigurable robot is demonstrated in [**17**]. Each “molecube” module is a 10-cm cube, and one half of it can swivel relative to the other half. Each half can bind with one additional module by using electromagnets. Lattice-based self-reconfigurability and self-replication of a four-module entity is also demonstrated in [**18**] when the system provides an ordered supply of additional units. The system executed a predetermined sequence of actions. ATRON is yet another lattice-based system, in which modules are arranged in a subset of a surface centered cubic lattice [**19**]. ATRON modules composed of two hemispheres joined by a single revolute joint, as shown in Figure 2.1. In [**20**], Brandt, Christensen and Lund discuss the mechanical design of ATRON and its resultant system properties, based on FEM analyses and real-world experiments. Fracta [**21**] and Metamorphic is also a homogeneous 2-D lattice-based mechanical hardware characterized by hexagonally shaped robot modules. Other lattice-based robots like 3-D SRS, I-Cube [**22**], and Proteo [**23**], are also homogeneous in nature, which provides for easy self-reconfiguration of these modules, but the hardware implementation is very complicated due to the geometric symmetry required for actuation and connection with other modules to provide more DOF's (Degrees Of Freedom).

Figure 2.1- Lattice type Architecture (Brandt et al., 2007)

2.2.2 Chain-based robot architecture

Chain-based architectures have units that are connected together in a string or tree topology. The chain or tree can fold up physically to fill arbitrarily shaped spaces, but the underlying architecture is still serial. Through articulation, chain architectures can potentially reach any point or orientation in space and are therefore more versatile than some other architectures, but computationally they are more difficult to represent and analyze, and therefore are more difficult to control. PolyBot [**14**], [**24**] is modular chain robot that can configure its shape without human assistance. Yim et al. [**3**], have explained the ability of PolyBots to self-reconfigure and self-reassemble with other

PolyBots despite the limitation of each PolyBot to a single DOF, as shown in Figure 2.2. CONRO [**25**], [**26**], [**27**] is a homogeneous modular chain robot with a processor, power supply, sensors, and actuators on each module. The CONRO robot has demonstrated the capability of self-assembly. M-TRAIN [**28**] is another modular, distributed, self-reconfigurable homogeneous robot module which can change configuration by changing positions and connections with other M-TRAIN modules.

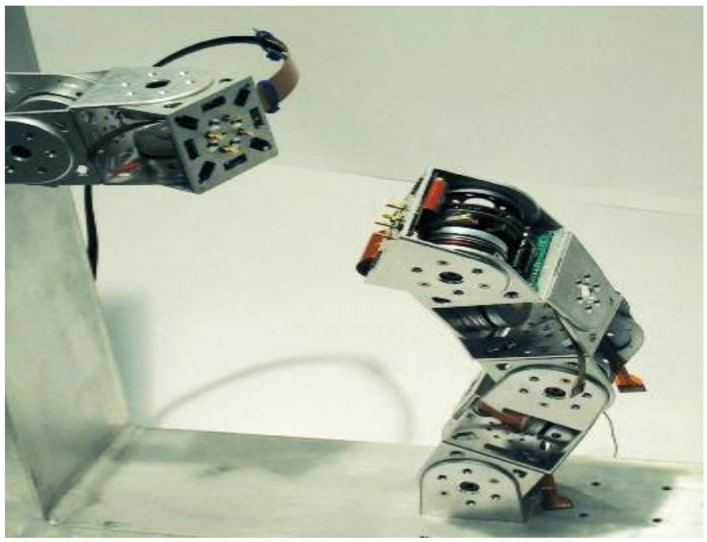


Figure 2.2 - Chain type Architecture (Yim et al., 2007)

(copyright @ 2007 Yim et al)

2.2.3 Mobile-based Architecture

Mobile architectures have units that use the environment to maneuver around and can either hook up to form complex chains or lattices, or form a number of smaller robots that execute coordinated movements and together form a larger “virtual” network. CEBOT [**29**], was proposed by Fukuda et al. with dynamically reconfigurable robotic systems and has heterogeneous modules with different functions. CEBOT has gone through great development, and the later versions are called CEBOT Mark 1, 2, 3, and 4 [**30**]. CYBOT [**31**] is another type of a medium-powered mobile robot that is cheap enough to mass produce and hence assemble an interacting swarm. Gupta [**32**], proposed a low cost mobile module, the AUTOBOT robot, which can estimate the distance of obstacles and recognize multiple robots in an environment. The AUTOBOT module is capable of performing short-range communication using a 2.4 GHz radio module and has two hours of battery backup power.

The S-BOT [**33**], is a fully autonomous small wheeled cylindrical robot, 12 cm in diameter, 19 cm high, weighing about 700 g, and is equipped with various sensors. S-BOT's mobility is ensured by a differential drive system and mobile robot attachment architecture capable of clinging to other S-bots similar to itself by using a gripper. Dorigo [**34**], has run set of experiments in which 18 S-bots demonstrated coordinated motion on rough terrain, hole and obstacle avoidance, self-assembly, cooperative transport, environmental exploration, and path formation. Recently, Swarm Bots (S-Bots) [**35**], have become one of the most popular swarm robot platforms because of their extreme plasticity, high degree of physical adaptation, and minimal need for human interaction and monitoring as shown in Figure 2.3. IROBOT is another popular platform often used for swarm research. McLurkin and Yamins [**36**], describe work in which researchers implement an algorithm on a group of 25 I Robot Swarm Bots and collect performance data. Each SwarmBot is mobile and has four IR transceivers at its corners, allowing communication with nearby robots and facilitating determination of the bearing, orientation, and ranges of its neighbors. A 32 bit micro processor is used as a controller and all robots are homogeneous. Red, Blue, Green LEDs and a MIDI audio system are used to provide audible and visual indications for monitoring the internal state of the robots.

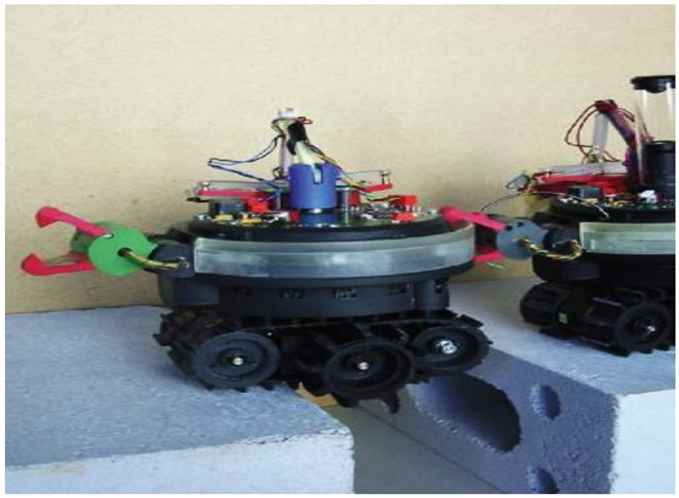


Figure 2.3 - Mobile type Architecture (Mondada et al., 2003)
(copyright @ 2003 Mondada et al)

Alice [**37**], [**38**], is a small rectangular mobile robot with dimensions of 22 × 21mm, driven by two high efficiency SWATCH motors for locomotion, controlled by a PIC16F877 microcontroller with 8K word of Flash EPROM program memory. Alice has four IR proximity sensors for obstacle detection, a short-range robot-to-robot communication system, and an IR receiver for remote control. Also, there are a wide variety of auxiliary modules for extending its abilities, such as a linear camera, RF, and gripper modules.

E-puck [**39**], is a circular robot with a diameter of 70mm, driven by two stepper motors for locomotion, controlled by a dsPIC 30F6014A microcontroller with 144KB of program memory and 8KB of RAM. E-Puck has eight IR sensors for measuring proximity to objects and for measuring ambient light. It has a speaker for audible feedback, three directional microphones that can be used for sound localization, and a 3-axis accelerometer. The robot has a color camera, a number of LEDs to signal/show its state, and Bluetooth for its main wireless communication channel. The robots can be programmed via the Bluetooth communication channel.

Table 2.1 below lists some self-reconfigurable robots, their classification and source of relevant reference information:

SYSTEM	CLASS	DOF	REFERENCE(s)
CEBOT	Mobile	Various	Fukuda et al. (1989)
Polypod	Chain	2	Yim(1993)
Molecule	Lattice	3	Chirikjian et al. (1996)
CONRO	Lattice	4	Kotay & Rus (1998)
Polybot	Chain	2	Castano et al. (2002)
Metamorphic	Chain	1	Golovinsky et al. (2004)
Telecube	Lattice	6	Suhet al. (2002)
I-Cube	Lattice	3	Unsal & Khosla (2001)
Pneumatic	Lattice	2	Inou et al. (2002)
Uni Rover	Mobile	2	Damoto et al. (2001)
M-TRAN	Hybrid	2	Murata et al. (2002)
Atron	Lattice	1	Brandt et al. (2007)
Swarm-bot	Mobile	3	Groß et al. (2006)
Superbot	Hybrid	3	Shen et al. (2006)
Molecube	Chain	1	Studer & Lipson (2006)
Miche	Lattice	0	Gilpin et al. (2008)
ACM	Chain	Various	Hirose & Mori (2004)
Miniturized	Hybrid	0	Tomita et al. (1999)
Fractum	Lattice	2	Yoshida et al. (1999)
M-TRAN II	Hybrid	2	Kurokawa et al. (2003)

Table 2.1 - Classification of swarm robots

Only a few systems include heterogeneous robots and such swarm systems have been found to be limited both physically and behaviorally. Table 2.2 below lists a number of swarm robot systems, along with their advantages and limiting factors.

Sr. No	System	References	Advantages and Disadvantages
1	PolyBot	Yim et al.	Advantages: 1st system to demonstrate the ability of self-reconfiguration with most active modules in a connected system. Each module fits within the 5cm cube. They are versatile in nature. Each module contains a Motorola PowerPC 555 processor with 1MByte of external RAM, and DC brushless motor with built in hall effect sensors. Limitation: Insufficient sensory unit for mapping of environment. Cannot work in unknown environment with rough surface or when obstacle avoidance is not possible.
2	M-TRAN	Yoshida et al.	Advantages: Very small actuated modules, highly-robust, miniature, and reliable. Quick self-reconfiguration and versatile robotic motion. Limitations: Connection mechanism works on an internally balanced magnetic field that is not strong enough to hold the other modules. Single M-TRAN module does not have enough DOFs for switching from one posture to another form. Lack of sensors leads to mapping and control problems. Power consumption is more as uses servo motor and electromechanical force for connectivity.
3	ATRON	Stoy et al.	Advantages: Each module is equipped with its own power supply, sensors and actuators, allowing each module to connect and communicate with a neighbor module. Able to sense the state of its connectivity and relative motion. Limitation: Since each module includes two-axis accelerometers only, a module cannot tell if it is turned upside down or not. When two modules are connected, it’s very difficult for them to move themselves, which requires cooperation from its neighbor. They are not mechanically stable and due to this mechanical instability, their electronic performance is poor.

4	SamBot	Hongxing Wei, et al.	Advantages: SamBot is a combination of mobile and chain-based modules capable of self-assembly and self-reconfiguration. SamBot uses 4 docking mechanisms for connecting with other SamBots. Detects other SamBots using Infrared sensors. Limitation: Infrared sensors limit the search range and require line-of-sight between SamBots. SamBot architecture lacks extra actuators, grippers, and sensors for gathering information about the working environment.
5	Swarm Bot (S-bot)	Mondada, et al.	Advantages: Robot swarms consisting of 2 to 40 S-bots have been successfully demonstrated. S-bots are fully autonomous mobile robots capable of self-navigation, perception of the environment and object. Capable of communicating other S-bots and transporting of heavy objects over very rough terrain. Limitations: Initial cost is high. Images and sound are the only way of communicating with other S-bots. Large number of sensors and actuators consumes power, reducing functionality and operating time.
6	CONRO	K. Stoy et al.	Advantages: Small, rectangular self-reconfigurable swarm robot with a low price, Versatile. Limitation: Uses onboard low-capacity batteries that limit the usefulness of modules. Limited sensors limit ability to sense surroundings. Only two controllable degrees of freedom.
7	MiLyBots	Luis Vega et al.	Advantages: Low-cost, reliable, robust, reusable, movable, size-efficient, power sparing, wireless, dynamically programmable swarm robots. Limitation: MiLyBots are not self-reconfigurable, self-assembled swarm robots. Lack actuators and connection mechanisms for physically attaching to other modules.
8	I – Cube	Unsal & Khosla et al.	Advantages: I−Cubes are low cost, small lattice based swarm robot with 3 DOF. Limitation: Unable to provide heavy object transport. Limited sensors. Lacks actuator mechanism.

Table 2.2 - Advantages and limitations of various robot platforms.

The swarm robot systems developed so far are confined to homogeneous hardware architectures, i.e. consisting of the same type of hardware structure and functionality, while only a few have been implemented as heterogeneous system. These homogeneous swarm robot platforms are limited in abilities and perform the same actions and tasks, which lead us to our first problem - are homogeneous swarm robots are effective enough for localization, mapping, search and rescue. Heterogeneous swarm robot platforms contain different capabilities and functionality, such as S-Bots, but the limitations with S-Bots include lack of diverse sensors, communication range, and misinterpreting range when the camera is reflecting off a spherical image. The hardware platform also has open issues in the swarm robotics research, such as heterogeneity, control mechanism, initial cost, size, shape, communication methods. Communication between each robot and localization of the swarm robots can increase the functionality and capability of the heterogeneous robot swarm, although there is no specific hardware or technique concluded in the swarm robot system.

2.3 Hardware Architecture Design Components

The hardware of robot swarms consists of a broad range of components, including a vast variety of sensors, actuators, controllers, cameras, etc. It is common practice to use customized hardware for specific applications, resulting in an increased degree of heterogeneity which in turn results in increased complexity for software developers. The nature of the tasks and the field of application influence the hardware architecture of a swarm robot, which must have the ability to navigate in dynamically changing environments without only third-party interaction, human or otherwise. The choice of appropriate sensors in robot swarms helps the individual robots to perceive the various physical properties of their surroundings. Based on measured data, the swarm robots may conclude that one or more particular action is necessary based on their current state. They will then activate and control actuator devices to interact with and influence their environment. In this section we review various hardware architectures for swarm robots based sensory platform, actuation, locomotion, controller, and power supply.

2.3.1 Sensors Platform Review

Sensors are used to provide information about the surrounding environment to the controller - a process known as mapping. In swarm robotics, sensors are used to detect obstacles, to find targets, to find paths, and for communication. There are many different types of sensors used in swarm robots, but the IR Proximity Sensor [**6**], [**14**], [**16**] is most commonly used. It is a small, cheap, easy to mount, and able to detect objects at a distance of 5cm to 15cm depending on the color of an object. Such an IR sensor is shown in Figure 2.4. An IR proximity sensor works by applying voltage to a pair of IR light-emitting diodes, in response to which they emit infrared light which propagates through the air. Once the emitted light hits or is blocked by an object, it reflects back to the sensor, the closer the object, the stronger the intensity of reflected light will be. Geunho [**40**], addressed practical design and hardware implementation of DRIr (Dual Rotating Infrared Sensor) proximity sensors for mobile robot swarms. These sensors are characterized by low cost, high reliability, and easy integratability into commercial mobile robots. The DRIr also provides robots with full 360 degree azimuth scanning and controllable range-tracking capabilities. Another type of sensor used in swarm robots is the Laser Range Finder (LRF) sensor, which has higher speed, accuracy, and resolution than LED-based IR sensors. LRF sensors have been used in various applications of mobile robots, but such applications are limited because of the high expense of LRF compared to other proximity sensing techniques [**41**]. Another type of proximity sensor is the Sonar or Ultrasonic sensor [**42**], [**43**] providing a mobile ultrasonic relative positioning system (URPS) that can be used by robots to detect the distances and angles of surrounding robots in relation to one another. Sonar time-of-flight distance sensor measurements work over a longer range than Infrared sensors, but can be easily affected by the hardness of objects, which can result in undesired measurement variation due to differences in how sonar waves are reflected and refracted by varying surface properties.

Figure 2.4 - IR Proximity Sensor Module

Some swarm robots use a vision system such as a camera to find out the position of other swarm robots as well for pathfinding and localization [**33**], [**34**]. The S-Bot (Swarm Bot) uses a VGA-resolution omni-directional camera for visual communication with other robotic units and to determine the position of a target for long and short distance sensing. LEDs of different colors are used for visual signaling with other robots. In some of the swarm robot modules [**44**], omni-directional microphones, humidity sensors, temperature sensors, axis accelerometers, incremental encoders, and torque sensors are used. Sometimes odometry sensors are also used to aid in exploring all the positions of swarm modules in a working environment.

2.3.2 Actuation and Locomotion Platform Review

The goal of a fully autonomous swarm robot team is to self-navigate, grasp objects, and physically interconnect with each other to accomplish self-reconfiguration, self-reassembly, and self-replication by means of a gripper or manipulator. Another goal is the transport of a heavy object from one location to another location in any type of terrain with the help of locomotion units such as wheels, tracks, treels (track/wheel combinations), or legs (quadrupedal, hexapedal etc.). Sensors and actuators must be selected and designed while considering constraints such as power consumption, voltage, driving signals (ideally pure digital), size and cost.

An artificial localization of swarm robots is mainly classified into two categories: absolute positioning and relative positioning [**22**]. In some swarm robots, a GPS system (Global Positioning System) is used to navigate in an unexplored environment. The GPS system consists of a number of satellites (originally 24, currently 32) in earth orbit, each transmitting time and position information that can be used with any receiver on or near the earth with an unobstructed view of at least four satellites to determine its position and altitude. The robot swarm can use the trilateration method to calculate absolute location to with a predetermined accuracy error. The accuracy error, the group deemed, isn't critically important since the robots can communicate with each other permitting them to determine the relative location with respect to each other. When they localize with each other, the searching algorithm allows the robot to cover more area with much more efficiency. However, as a GPS system for determining absolute position is relatively expensive, another simple localization technique known as odometry is commonly used. This technique is accurate in the short-term and inexpensive. This technique uses wheel revolution data to find out linear displacement relative to the floor. The drawback to this technique is that, it is highly sensitive to error; that is, if there is a slight error in calculation, then the entire set of location calculations is skewed. The Servo motors are used for the locomotion in the swarm robots in addition with an incremental encoder or odometry unit. The actuation modules are of the following types:

2.3.2.1 Wheeled Swarm Robot

This swarm robot module might have two wheels for locomotion driven by servo motors. Most mobile robots only provide simple motion control by switching the DC servo motors on and off. E puck [**39**], Alice [**38**], and Sumobot use a 2-wheel robot module, while SamBot [**33**] is maneuvered by means of a multi-crawler robot created by self-assembly. The three-wheel [**31**] and Boe bot platforms are also used in swarm robots, with gear assembly attached to a DC motor. The shape of the platform might be triangular or circular. Between 1995 and 1997, Takeshi Aoki, Yuki Murayama and Shigeo Hirose, [**45**] built an omni-directional three-wheel planetary exploration robot, the Tri-Star. The chassis is deployed at the exit of the container and the wheels are expandable.

Some swarms use four wheels for movement and locomotion. The omni-directional mobile robot described in [**46**] is equipped with four independent driving wheels equally spaced at 90 degrees from one another. The drawback of having a wheeled robot is that, if any obstacle comes in the way of the robot, the robot may not be able to run over that obstacle. Also, the speed of a wheeled robot changes with changes in surface roughness and inclination. However, wheeled robots require little power and are energy efficient.

2.3.2.2 Tracked Swarm Robot

Tracked robots use crawl units or tracks similar to those used for terrestrial mobile applications like military tanks and automobiles. These tracks are especially suited for motion on difficult terrain. The robot Aurora Automatika in Pennsylvania, built by Hagen Schempf in 1999, consists of a single and directional track. The University of Wuerzburg built a two-tracked Nanokhod robot, with an articulated pendulum used as a weight-cons and itself made of a caterpillar. It can move horizontally on slopes. The Nanokhod [**47**], is a miniaturized track-enabled robot that was developed based on Russian technology. The tracker consists of two “caterpillar” track units, a tether unit, and a payload cabin. The caterpillar tracks are driven by four internal drive units, each consisting of a stepper motor attached to a 64:1 planetary gear in front of a crown and pinion stage. The output stage is a miniaturized harmonic drive whose input is coupled directly to the crown gear. The omni-directional mobile robot is equipped with four independent driving wheels equally spaced at 90 degrees from one another. The tracked robot has better traction capability on loose soil and can handle large hinder and small holes, but it is inefficient because of the friction of tracks that “scrub” along surfaces while turning.

2.3.2.3 Leg-based Robot

Some swarm robots use legs for locomotion, but they are very complex to build and controlling the legs is also complicated. They tend to be very slow and create an impact with each step.

2.3.2.4 Hybrid Robot

A main premise behind hybrid robot architecture is that the combination of any two mechanisms is better than a single one, as it benefits from the advantages of the two. This concept is highly used in recent prototypes such as the Swarm Bot [**34**] and S-Bot [**35**]. The S-Bot is based on track and wheel combination platforms called Treels, as shown in Figures 2.5 and 2.6. Each treel is controlled by an independent motor so that the s-bot can freely move in any environment and can easily rotate on a spot. This mechanism allows each s-bot to move over moderately rough terrain with complex obstacles. AutoBot [**32**] uses a differential drive with reliable motion control configured with caster wheels and a pulse width modulation technique that is employed for DC motor control.

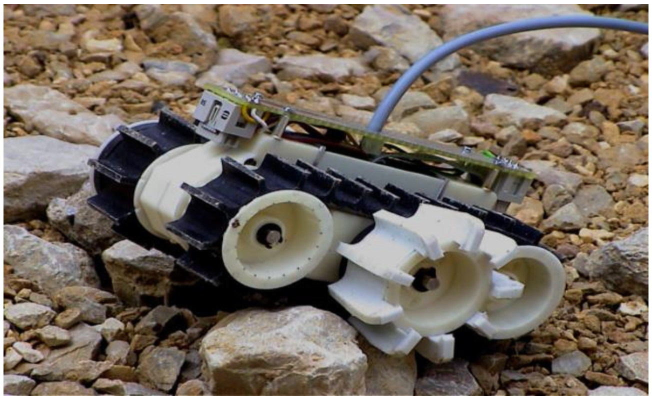


Figure 2.5 - S-Bot Tracks (Francesco Mondada et al, 2002)

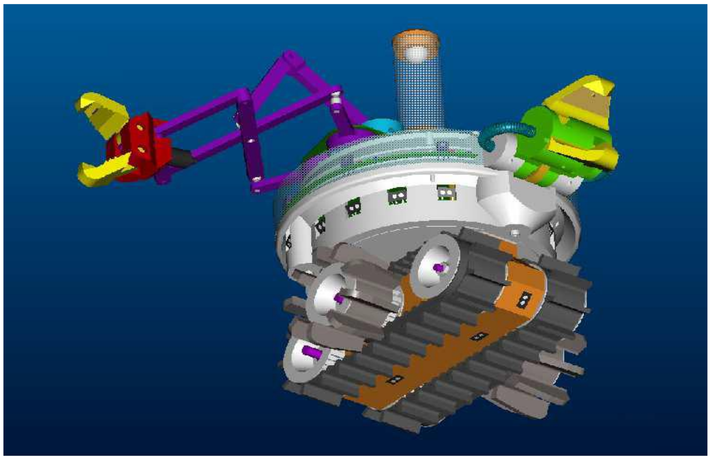


Figure 2.6 - Bottom view of the s-bot robot showing its independently-controlled treels (Mondada et al, 2002) (copyright @ 2002 Mondada et al)

Piezoelectric actuators are commonly used for locomotion and actuation of mobile micro robots of a size between 1 dm^3 and 1 cm^3. Figure 2.7 shows the Pioneer II P2AT-8 robots and custom-built track robot at ACE Lab, UTSA. In this module, a sonar based sensor platform is used with both four-wheeled robots and tracked robots.

Figure 2. 7 - Pioneer II P2AT-8 (copyright @ 2007 Pioneer Inc., et al)

Manipulation of objects by swarm robots is accomplished by grasping, pushing, and caging [**48**]. Grasping action includes form closure and force closure techniques. By way of example, grippers [**34**], [**35**], [**47**] are used as manipulators on almost all swarm robots, both for interconnection with other swarm robots and for grabbing (grasping) objects. Such grippers are usually operated by a DC motor. Opening and closing of the grippers are typically measured by an optical sensor. In Voyles' Terminator [**49**], the robot consists of a cylindrical body with dual 3-degree-of-freedom (DoF) arms controlled by two gear motors that can be fully stowed inside the body of the robot. Some robots use a mechanical hook controlled by a spring return actuator to connect with other robots. Minghui et al. [**50**] describes a wheel-manipulator robot consisting of a triangular wheel and a 5-DOF arm with an end-effector. Other connection techniques include point-to-point and surface-to-surface attachment mechanisms. M-TRAN, CORNO and I-CUBE [**19**] use a surface-to-surface connection in which an active attachment-making connector extends three hooks from it's mating surface to grab onto features of a passive mating connector. The passive connectors are built from two bars of stainless steel rigidly integrated in the hemisphere. These three hooks are driven by a DC motor via a worm gear. MiLyBot [**51**] uses a Solarbotics motor, as shown in Figure 2.8. These motors exhibit low power consumption and excellent torque.

Figure 2.8 - Solarbotics motor and gearbox assembly used in MiLyBot (Luis Vegas, et al, 2008) *(copyright @ 2008 Luis Vegas et al)*

2.3.3 Controller and Communication Module Platform Review

In robot swarms, communication can work using any of the several different techniques, depending upon factors like robot size, robot cost, budget, the environment in which the robots will work, and other application-specific limitations. Generally speaking, swarm robots are controlled by one of two broad approaches: a centralized approach where a single supervisory robot plans for the group, or a distributed approach where each robot is responsible for its own planning [**52**]. In recent years, robots have become more mobile, requiring wireless communication techniques like Bluetooth, wireless LAN, stigmergy, or visual signaling using IR LED's. "Stigmergy" was introduced by Pierre-Paul Grasse in the late 1950's to describe the type of indirect communication employed by insect life, such as ants and termites, using pheromones to mark the shortest path back to the nest, to mark the location of food sources or to identify danger. However, communication can also be established by sending messages to other robots using Bluetooth, wireless LAN or infrared LEDs. Infrared LEDs are used in SWARM BOT; S-BOT achieves visual communication with different color LEDs and a camera mounted on the top of robot to receive signals from other robots. This technique is very economical and easy to install on minirobots or swarm robots, but sunlight or other light sources can interfere with this type of communication. Wireless LANs may used on midsized robots to send high-volume message traffic to the robot team, but this type of network can be disturbed by other RF-radiating devices. One of the best and most cost-effective techniques for communicating with team swarms is Bluetooth communications, which needs a unique ID for each swarm robot. There are many Bluetooth devices or cards available on the market, using both WaveLAN and Bluetooth wireless communication systems with wireless antennas.

Ming et al. [**53**] explored the use of wireless mesh networks (WMNs) and mobile ad hoc networks (MANETs) for robot communication with the help of mesh routers, PDA's, wireless adaptors and GPS' on each robot. In CORNO [**12**], each module communicates with other modules by an IR transmitter and receiver to form a local communication network. CORNO is controlled by a BASIC STAMP 2 processor card. M-TRAIN [**28**], also uses the BASIC STAMP 2 microcontroller for controlling the modules and each module communicating through the Relay PIC by using serial communication. AUTOBOT [**32**] uses 2.4 GHz, 1 Mbps GFSK radio-based local communication by a Cypress CyFi™ CYRF7936 radio integrated circuit and an integrated PCB Trace antenna. A swarm robotics project by Samanta et al [**54**], at

Villanova University, PA employed LEGO NXT mobile robots with Bluetooth communication via NXT bricks each containing an Atmel 32-bit ARM7 processor running at 48 MHz with 64 KB of RAM and 256 KB bytes flash memory. Communication between the NXT robots and a PC laptop host was implemented using a D-Link DBT-120 wireless Bluetooth 2.0 USB Adaptor.

The Autonomous Miniature mobile Robot (AMiR) [**55**] uses an AVR microcontroller series as the main processor for managing all AMiR's modules. This microcontroller, an ATMEGA168 clocked at 8.0 MHz, using its internal RC oscillator and IR-based local communication for each module. At the Cleveland State University [**56**], the square robot swarm was designed, communication between these robots and the base station is accomplished using a MaxStream 9Xtend RF transmitter/receiver and the PIC18F4520 a microcontroller with the C language compiler. KOBOT [**57**] was designed as a self-organized flocking robot using an IEEE 802.15.4/ZigBee compliant XBee wireless module with a range of roughly 20 m for communication between robots, a PIC 18F4620A microcontroller, and a PC (supervisor). There are also other commercially available low-cost microcontroller devices available such as the Arduino, which is a flexible and open source electronics platform, easy to use and easy to program in various programming languages. Wireless communication can also be established using CC250 with Atmega16 built-in Universal Asynchronous Receiver Transmitter (UART).

2.3.4 Power Option

Another important consideration for swarm robots is the power supply, since each swarm robot is very small and mobile in nature, suggesting the power supply should be small and light enough to be mounted on the robot. Most swarm robots work on 5 to 25 V DC power supplied by rechargeable lithium batteries. Lithium-Polymer batteries (Li-Po) [**58**], have several advantages in such applications, including: high energy density, thin size, and operational safety when compared with other rechargeable batteries. The ATRON swarm robot module, 11cm in diameter, equips each module with two 3.6 V 980mAh ion-lithium-polymer cells. This provides 7.2 volts at an ampacity of 980mAh for each module. The S - bot is equipped with two Lithium-ION batteries placed between the tracks. The power storage capacity of these batteries is 10Wh. Preliminary measurements show a power consumption of for one S-bot, between 3 and 5W, which ensures continuous operation for at least two hours.

AutoBot is powered by an 11.1v Li-Po battery with 500mAh ampacity. CYBOTS are smaller, around 25cm in length, and use a pair of Li-Po rechargeable battery as a power source.

CHAPTER 3: RESEARCH MODEL

3.1 Introduction

Currently, most existing swarm robot systems have been designed and implemented with homogeneous hardware. Only a few of them have heterogeneous robots and those swarm systems were limited physically and behaviorally. Due to the lack of methods and tools, swarm robot designer cannot achieve complexity required of the real world applications [**59**]. In our review article [**60**], we have reviewed the hardware architecture of the robot swarm with self-configurability, self-assembly, and self-replication. After reviewing existing swarm systems and studying the limitations, we decided to design and built our own robot swarm system. In this design we have considered some important factors such as it's size, cost, autonomy, flexibility, robustness, power consumption, weight, etc. The main goal of our research is to build a heterogeneous robot swarm system in which each robot has a distinctive hardware compared with other robots.

Nowadays, electronic products are cheaper, smaller, lighter in weight and easily available, which makes robot swarms more cost-efficient, lighter in weight, and compact in size [**61**], [**62**]. Table 3.1 summarizes the hardware platforms implemented so far in swarm robot research experiments.

Sr. N0.	**Name**	**Sensor**	**Actuation**	**Controller**	**Communication**	**Positioning system**
1	E Puck	11 IR, Contact ring, Color camera	wheeled	dsPIC	Bluetooth	Expansion IR based
2	Alice	IR, Light Sensor, Linear Camera	wheeled	Microchip PIC	Radio (115 kbit/s)	------
3	Jasmine	8 IR	wheeled	2 ATMega	IR	Integrated IR based
4	I-swam	Solar cell	3 micro leg piezoelectric actuator	Not Available	Not Available	-------
5	Khepera	8 IR	wheeled	Motorola MC66831	RS232, Wired link	-------
6	Khepera III	11 IR, 5 Ultrasound	wheeled	PXA-255, Linux, dsPIC	WiFi & Bluetooth	Expansion IR based
7	S-Bot	15 Proximity, Omnidirectional Camera, Microphone, Temperature	Wheeled, 2 gripper	Xscale Linux PICs	WiFi	Camera based
8	SwarmBot	IR, Camera, Light, Contact	Wheeled	ARM and FPGA 200 kgate	IR based	Integrated IR based
9	Kobot	8 IR, Color camera	wheeled	PXA-255, PICs	ZigBee	Integrated IR based

Table 3.1 - Previously developed hardware platform summary in detail

The hardware platforms given in Table 3.1 are homogeneous in nature and limited with capabilities and functionality. To implement the UB robot swarm system we have used $2,040 from our allocated budget of $2,500 to build five robots. The size and cost of each robot in the UB swarm is given in Table 3.2.

Sr No	**Name**	**Size (mm)**	**Cost ($)**	**Images**
1	Rover 1	225 x 245 x 84	$ 325.00	See fig. 5.6
2	Rover 2	200 x 108x 58	$ 300.00	See fig. 5.7
3	Rover 3	160 x 165 x 95	$ 500.00	See fig. 5.8
4	Rover 4	135 x 100 x 95	$ 500.00	See fig. 5.9
5	Rover 5	135 x 100 x 85	$ 315.00	See fig. 5.10

Table 3.2 - UB Swarm Robots with their Size and Cost

3.2 Hardware Design

The hardware design for any swarm is an interactive and an important phase; as all components and/or parts are assembled to build one robot swarm. At the hardware level, the most work has been done in collective behavior with homogeneous robots. In this proposed architecture, we decided to exploit reconfigurability, and modularity using heterogeneous robots with centralizes and/or decentralized control algorithms which are influenced by ants, bee colony, and insect’s behaviors. This modular hardware architecture consists of independent sensory units, actuator module, communication unit, allowing the swarm system to be extendible, flexible.

The UB robot swarm is simple, capable of sensing, localization, and actuation based on the local information and basic rules. In the following sections, the mechanical and electronic modules of the robots are described with their working. All the parts were tested and slightly modified for the applications, and then assembled to build the physical robots swarm. The software scans for replacing or extra added sensors itself, which makes robot swarms more dynamic.

There are many things that have to be considered when designing the hardware platforms for the heterogeneous robots such as each module should be easily modifiable and compatible with a high performance microcontroller. They should be reconfigurable and provide the easy support for the software as well as for the middleware.

3.3 Sensory Platform

Gathering information or data about the working environment or surrounding environment of the swarm robots is an everlasting job. The sensory unit is essential for robot swarms to perform the tasks such as obstacle detection, obstacle avoidance, detecting its neighboring robot, and navigation. Sensors are classified as five sensing elements of the robot swarm and are used to collect the information about their surrounding environment by electrical or electromechanical signals. In this proposed hardware design, each robot swarm is equipped with different types of sensors such as temperature sensor, humidity sensors, encoder, camera, communication devices, proximity sensors, ranger detector, GPS tracking devices, etc. There are two primary factors that affect the limitation of sensors, they are

- Range and resolution of the sensors.
- Noise that affects the output of the sensors.

The study of animal behavior shows that, the sensory skills are developed and adapted by the interpretation of signals produced from sensors. In swarm robots, this self-learning ability is achieved by configuring and calibrating sensors for a given task. Using multiple sensors provides the most efficient and effective methods for collecting, and exploring the unknown environments. In this section we have explained all the sensors that are used in our proposed robot swarm hardware with their technical specifications.

3.3.1 Proximity Sensors

Distance measurement and obstacle avoidance are the fundamental element of the information gathering quest. In swarm robotics, obstacle detection and collision avoidance in real time while the robots are in motion is major constraint and difficult task. Proximity sensors sense the object or surrounding material or other moving swarm robots without any physical contact, and calculate or give very precise distance of that object [**63**]. This crucial component not only avoids collisions, but also prevents the physical damage to the swarm robots and maintains safe distance [**64**]. Depending on the type of technology used, proximity sensors are classified into different categories such as, inductive, capacitive, photoelectric, and ultrasonic proximity sensors.

Among these, ultrasonic proximity sensors were found to be more accurate and have more capabilities when compared to the other types of proximity sensors. In proposing a swarm robot model, we use ultrasonic as well as photoelectric (Infrared) proximity sensors.

3.3.1.1 Ultrasonic Proximity Sensor

Ultrasonic sensors are commonly used to measure distance because they are inexpensive and easy to handle. They are used to avoid obstacles, to navigate, and for map building. Ultrasonic sensor emits sound waves (ultrasound) of the 20 KHz frequency and uses it to find a way around the obstacle, detect the uneven surfaces, any shape and size of object in known as well as in unknown environment. This is known as Echolocation. This sensor sends out ultrasonic waves which are then detected after they are reflected or bounced back from the object and/or obstacle. The time required for sending and to receiving the ultrasonic waves are measured and further processed to calculate the distance. These sensors are very precise in measurement and used in applications that require measurement between stationary and moving objects. In our proposed hardware architecture design, ultrasonic sensors are mounted on the sides (left and right), front and back corners of the robot. Following are the ultrasonic sensors used in a UB robot swarm system with their technical specification.

A. Devantech SRF02

The SRF02 is a single transducer; low cost, high performance ultrasonic range finder as shown in figure 3.1, which works in two modes - I2C and a Serial interface mode. Following are the specification of sensor:

SRF02 Specification:

- Voltage - 5v only required
- Current - 4mA Typically

- Frequency – 40 KHz
- Range - 15cm - 6m.
- Analogue Gain - Automatic 64 step gain control
- Connection Modes –

  1 - Standard I2C Bus.

  2 - Serial Bus - connects up to 16 devices to any uP or UART serial port
- Full Automatic Tuning - No calibration, just power up and go
- Timing - Fully timed echo, freeing host controller of task.
- Units - Range reported in uS, mm or inches.
- Light Weight - 4.6gm
- Small Size - 24mm x 20mm x 17mm height.

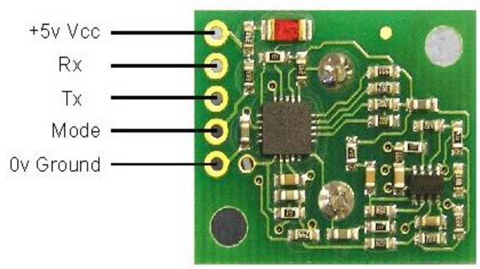


Figure 3.1- Devantech SRF02 ultrasonic sensor

We are using the SRF02 in Serial mode and to use in serial mode, the mode pin is connected to 0v Ground. The Rx pin is data into the SRF02 and connected to the Tx pin on PIC controller. The Tx pin is data out of the SRF02 and connected to the Rx pin on PIC controller. As we are using multiple SRF02's, we connected them all up to the same serial port on PIC controller. The Tx from controller is connected to all the Rx pins on the SRF02's and the Rx pin on controller is connected to all the Tx pins on the SRF02's. This works because the Tx pins have high impedance (just a weak pull-up to 5v), except when actually sending data. The beam width pattern for detection of the object is shown in figure 3.2.

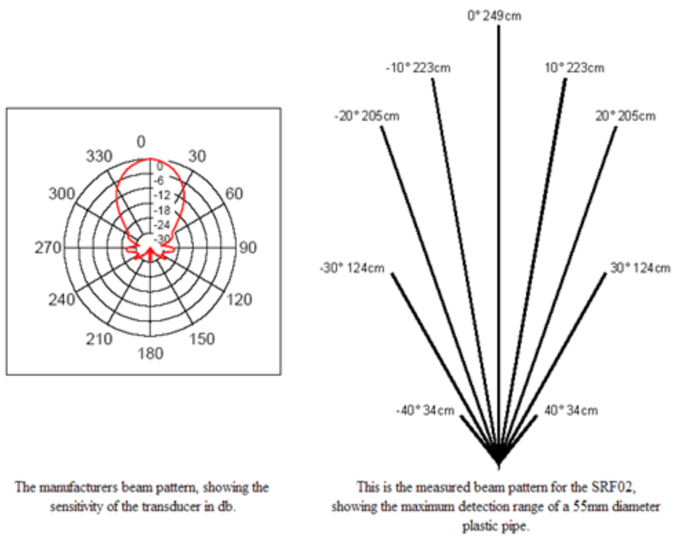


Figure 3.2 - Beam pattern and detection range

A. Seeedstudio Ultrasonic Range Finder

Seeed ultrasonic sensor as shown in figure 3.3 is noncontact distance measurement module with industrial performance.

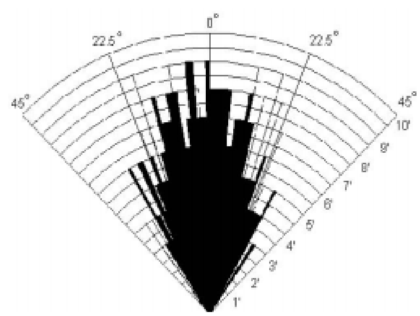
Figure 3.3 - Seeedstudio ultrasonic range finder

We are using the SRF02 in Serial mode and to use in serial mode, the mode pin is connected to 0v Ground. The Rx pin is data into the SRF02 and connected to the Tx pin on PIC controller. The Tx pin is data out of the SRF02 and connected to the Rx pin on PIC controller. As we are using multiple SRF02's, we connected them all up to the same serial port on PIC controller. The Tx from the controller is connected to all the Rx pins on the SRF02's and the Rx pin on the controller is connected to all the Tx pins on the SRF02's. This works because the Tx pins have high impedance (just a weak pull-up to 5v), except when actually sending data. The beam width pattern for detection of the object is shown in figure 3.2.

Distance in cm = pulse width / 58

The specification and the beam width pattern are shown in figure 3.4.

Supply voltage	5 v
Global Current Consumption	15 mA
Ultrasonic Frequency	40k Hz
Maximal Range	*400 cm*
Minimal Range	3 cm
Resolution	1 cm
Trigger Pulse Width	10 μs
Outline Dimension	43x20x15 mm



Practical test of performance, Best in 30 degree angle

Figure 3.4 - The specification and beam pattern of Seeedstudio ultrasonic sensor

B. Ping Ultrasonic Sensor

This sensor as shown in figure 3.5 is perfect for the applications which require measurement between the moving or stationary objects. The output from the PING sensor is a variable-width pulse that corresponds to the distance to target. The technical specifications of the PING sensor are given below;

Technical specifications:

- Range - 2cm to 3m (~.75" to 10')
- Supply Voltage: 5V +/-10% (Absolute: Minimum 4.5V, Maximum 6V)
- Supply Current: 25 mA typ; 30 mA max
- 3-pin contact (power, ground, signal)
- 20 mA power consumption
- Narrow acceptance angle
- Simple pulse in/ pulse out communication
- Indicator LED shows measurement in progress
- Input Trigger - positive TTL pulse, 2 uS min, 5 uS typ.
- Echo Pulse - positive TTL pulse, 115 uS to 18.5 mS
- Echo Hold-off - 350 uS from fall of Trigger pulse
- Burst Frequency - 40 kHz for 200 uS
- Size - 22 mm H x. 46 mm W x. 16 mm D (0.85 in x. 1.8 in x. 0.6 in)

The GND pin is connected to the GND of the microcontroller, 5 V is connected to the 5 VDC power supply and the signal pin is connected to the analogue pin of the micro controller.

Figure 3.5 - Ping ultrasonic sensor

C. URM37 V3.2 Ultrasonic Sensor

URM37 V3.2 Ultrasonic Sensor as shown in figure 3.6 uses an industrial level AVR processor as the main processing unit with a temperature correction which is very unique in its class. The technical specifications are given below,

- Power: +5V
- Current: <20mA
- Working temperature: -10℃～+70℃
- Detecting range: 4cm-5m
- Resolution: 1cm
- Interface: RS232 (TTL), PWM
- Servo control: One servo control output
- Operating Mode: Serial; PWM mode; Autonomous Mode; On/OFF Mode
- Temperature sensor: 12 bits reading from serial port
- Size: 22mm × 51 mm
- Weight: 30g

Figure 3.6 - URM37 V3.2 ultrasonic sensor

The pin configuration is given below:

Pin 1 - +VCC - +5V Power

Pin 2 - GND – Ground

Pin 3 - RST – Reset

Pin 4- PWM - PWM Output 0—25000US， Every 50 uS represent 1cm

Pin 5 - MOTO - Servo control signal output

Pin 6 - COMP/TRIG - COMP - On/OFF mode; TRIG - PWM or RS232 trigger

Pin 7 – NC

Pin 8 - RXD -RS232, TTL communication

Pin 9 - TXD - RS232, TTL communication.

D. LV-MaxSonar-EZ1 MB1010 Sensor

The LV-MaxSonar-EZ1 as shown in figure 3.7 is used where sensitivity needed along with the side object rejection. The LV-MaxSonar-EZ1 is the low cost, high quality ultrasonic distance sensors which offer easy to use outputs, no sensor dead zone, calibrated beam patterns, stable range readings, low power demands, and a host of other following features.

- Continuously variable gain for beam control and side lobe suppression
- Object detection includes zero range objects
- 2.5V to 5.5V supply with 2mA typical current draw
- Readings can occur up to every 50mS, (20-Hz rate)
- Free run operation can continuously measure and output range information
- Triggered operation provides the range reading as desired
- All interfaces are active simultaneously
- Serial, 0 to Vcc, 9600Baud, 81N
- Analogue, (Vcc/512)/ inch
- Pulse width, (147uS/inch)
- Learns ring down pattern when commanded to start ranging
- Designed for protected indoor environments
- Sensor works at 42 KHz
- High output square wave sensor drive (double Vcc)

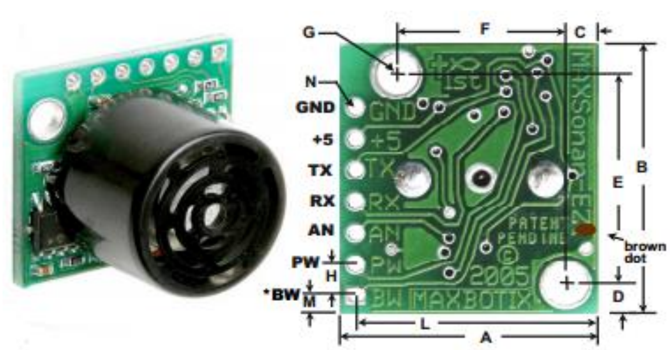


Figure 3.7 - LV MaxSonar EZ1 MB1010 sensor

The analogue pin of the sensor is connected to the analogue pin of the controller. The analogue voltage pin outputs a voltage which corresponds to the distance. The distance of an object from the sensor is directly proportional to the voltage. The sensor is designed to report the range to the closest detectable object. The range of the sensor can be calculated as given below,

First we need to do the scaling,

$$Vi = (Vcc/512)$$

Where,

Vcc = Supplied Voltage

Vi = Volts per inch (Scaling)

Once you know the voltage scaling, then you can calculate the range,

$$Ri = (Vm/Vi)$$

Where,

Vm = Measured Voltage

Vi = Volts per Inch (Scaling)

Ri = Range in inches

3.3.1.2 Infrared Sensor

The IR Range Finder works by the process of triangulation. A light pulse of wavelength range 850 nm (+/-70nm) is emitted from the sensor and then reflected back by an object or not reflected at all. When the light returns, it comes back at an angle that is dependent on the distance of the reflecting object as shown in figure 3.8. Triangulation works by detecting this reflected beam angle and by knowing the angle, the distance can then be determined. The performance of the IR sensor is limited by its poor tolerance to the ambient light or bright object color reflection. The IR range finder receiver has a special precision lens that transmits the reflected light onto an enclosed linear CCD array based on the triangulation angle. The CCD array, then determines the angle and causes the range finder to then give a corresponding analogue value to be read by the microcontroller. The output of the IR sensors is analogue, which is connected to the analogue pin of the microcontroller.

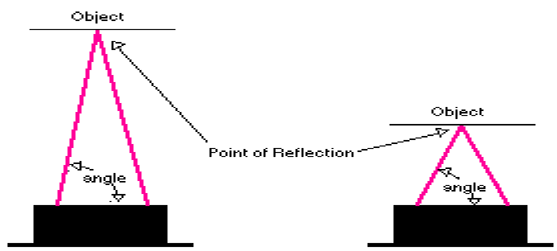


Figure 3.8 - IR Sensor triangulation process

A. Sharp IR Range Finder

GP2Y0A02YK0F as shown in figure 3.9 is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IRED (infrared emitting diode) and signal processing circuit. The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method. This device outputs the voltage corresponding to the detection distance.

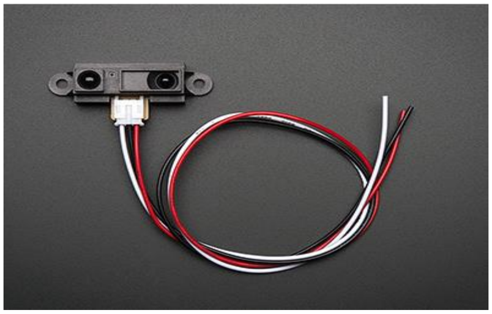


Figure 3.9 - Sharp IR range finder GP2Y0A02YK0F

B. Dagu compound infrared sensor

This sensor as shown in figure 3.10 is composed of four IR sensors which allow determining the location of an object edge more accurate and easily than single sensor. The IR diode goes on and off, emits the infrared light and the reflected IR light from an object is received by the IR sensor. By measuring the level of infrared light we can calculate the distance. The four analogue outputs from IR sensor are connected to the analogue inputs of microcontroller to measure the distance and avoid obstacle.

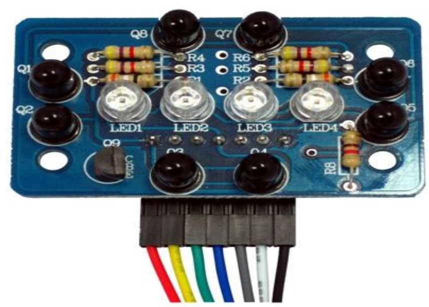


Figure 3.10 - Dagu compound IR sensor

3.3.2. Humidity and Temperature Sensor SHT1x

We are using fully calibrated digital SHT1 humidity and temperature sensor as shown in figure 3.11, mounted on a small PCB, integrated with signal processing unit. The sensor uses CMOS technology, which guarantees excellent reliability and long-term stability. The two wire serial interface and internal voltage regulation provide easy and fast integration with any microcontroller. This sensor consumes very low power and can be triggered only when needed.

Figure 3.11 - Humidity and temperature sensor SHT1

3.3.3. Encoder

Odometry is a reliable, precise technique and inexpensive technique to determine the exact position or location of the robots. Encoder counts the number of pulses for every rotation of the wheel, which then calculates the distance. The encoder has the IR reflective sensors which read the black and white strips on the encoder wheel which is attached to the shaft and the sensor unit is mounted on the chassis. When the shaft starts rotating, the encoder wheel also rotates and the sensor board counts the revolutions [**65**]. The encoder shown in figure 3.12 is mounted on the chassis with micro metal gear motor. This encoder has two IR reflective sensors with a phase difference of 90 degrees and the lead - lag of the waveform will decide the forward and reverse rotation of the wheel. This encoder works on 3.3 - 5 VDC voltage and the pulse output is 48 per revolution.

Figure 3.12 - Mini robot chassis encoder (SKU: SEN0116)

The Rover 5 uses a quadrature incremental encoder which measures the speed and the direction of a rotating shaft. The encoder has white (A) and yellow (B) wires which are used to measure the speed and direction of the robot. These wires are connected to the PIC controller, digital input pins and the red wire is connected to the power pin and black wire is connected to the ground pin. The encoders are mostly used for mapping application for localization of the robot using odometry. As the circumference of the wheel is known, we can calculate how far the robot moved from its previous position using the encoder reading. The reading, which we got from the encoder is, always has an error due to slippery action of the wheel with surface or ground and this error called as slip error. We have to subtract this slip error factor from our calculated reading to get an exact location. For getting exact value we have implemented the formula in our program. In [**66**], have proposed and tested the benchmark methods for all types of the errors in odometry.

3.3.4 GPS/GPRS/GSM Module

Solving a task which is beyond the capability of the single robot requires cooperation from the other swarm robots. For such a cooperative task, robots must communicate with each other and know their relative position and orientation [**67**]. The proposed swarm system is self-reconfigurable and heterogeneous. To achieve the heterogeneity one of the robot using the GPS/GPRS/GSM module shield as shown in figure 3.13, while other robots are using encoders, vision navigation to send its relative position to the other robots as well as to the host computer.

Figure 3.13 - GPS/GPRS/GSM module (© Robotshop INC)

Following are the technical specification of the module,

- Power supply: 6-12v
- Low power consumption (100mA@7v - GSM mode)
- Quad-Band 850/900/18001900MHz
- GPRS multi-slot class 10
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
- Class 4 (2 W @ 850/900 MHz)
- Class 1 (1 W @ 1800/1900MHz)
- Control via AT commands(GSM07.07, 07.05 and SIMCOM enhanced AT Commands)
- Directly support 4*4 button pad
- Buzzer for call notification
- LED indicators for power supply, network states and working modes

This module is compact and all in one solution for navigation and localization as well also saves significantly in time and cost for the integration of external added hardware parts. The GPS localization and navigation is mostly used for outdoor application and it's very efficient and accurate.

3.3.5 Camera Module

The camera module provides vision based localization and obstacle avoidance in the swarm system. We use Blackfin Camera with Radio/Motor Board as shown in figure 3.14, on our robot swarm. This camera can transmit the live feed to the host computer over wireless communication. In distinguishing between the obstacle and goal objects, IR sensor and ultrasonic sensor have some limitations, which can be rectified by using the camera module. We can view the images on the host computer or we can also feed them to the microcontroller with the onboard image processing unit. This camera is mounted on the SRV1 platform and DF robot rover platform.

Technical specification:

- 500MHz Analog Devices Blackfin BF537 Processor (1000 integer MIPS)
- 32MB SDRAM, 4MB SPI Flash
- SPI Flash and UART boot mode select
- External I/O Header (32-pin - 16 x. 2 x. 0.1")
- 3.3V Input - 145mA total draw at 500MHz, including camera
- Board dimensions - 50 mm x 60 mm (2.0" x 2.6"), 36g (1.25 oz) including camera
- UARTS - tested at up to 2.5Mbps with CTS/RTS flow control
- Timers (2 share pins with UART1)
- SPI - 2 slave select, 1 master select
- I2C

- 16 GPIO
- RoHS compliant
- Omnivision OV7725 VGA low-light sensor or OV9655 1.3 megapixel sensor
- Radio/Motor Control Module
- WiFi communication via Lantronix Matchport WLAN 802.11g radio
- On-board 3.3V high efficiency switching regulator (Recom R-783.3-1.0) for battery input (4.75 - 18.0 VDC)
- Dual H-bridge motor driver (Fairchild FAN8200) with 1000mA capacity per cycle.

Figure 3.14 - Blackfin camera module

3.4. Locomotion and Manipulation

The biggest challenges in developing the robot swarm lie in making them mobile, fully autonomous and versatile in order to move from one place to another over different types of terrains in an unknown environment. The locomotion of robot can be achieved by the motors with some gear ratio to slow down the speed of rotation and increase the torque. In manipulation, objects are moved from one place to another with the help of actuators as well as to rotate the wrist or open and close the gripper to grab the objects. In our previous work, the locomotion and manipulation of different robot platforms is explained in detail. In this section, we explain the type of motors used and their connection and control mechanism with microcontroller. The robot swarm uses track and wheel for locomotion and for manipulation uses robot arm which are driven by the DC motors, Geared DC motors, and Servo Motors. These motors need motor controller to control their speed of rotation and the direction. The number of rotations can be measured by the encoder to determine the exact position of the robots using odometry.

The drive motor is selected based on the voltage, RPM, brushed or brushless parameters. The UB swarm robots are driven by motors which are attached to the wheel. On each robot, two motor are attached to the wheels along with encoder modules. We are using DC gear motors; Solarbotics gear motors, Micro-metal gear motors, Tamiya gearbox motors. These motors are actuated and controlled using the motor controllers.

3.4.1. Tamiya Twin-Motor Gear Box

The twin gear motor box used two FA-130 motors with plastic gear assembly to provide speed and power to rotate the hex shaft. The figure 3.15 shows the Tamiya Twin-Motor Gearbox with shaft. Following are the technical specification of the Twin motor gearbox:

- Gear Ratios: 58:1 207:1
- Motor: FA-130
- Motor RPM: 12300 (9710 Maximum Efficiency)
- Motor Voltage: 1.5-3V (1.5V Recommended)
- Motor Stall Current: 2.1A
- Free-run current: 150mA

- Motor Stall torque: 36 g-cm

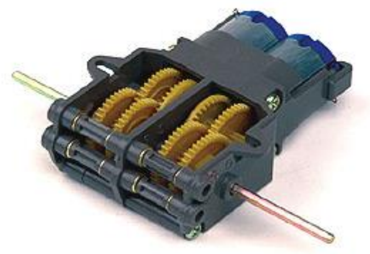


Figure 3.15 - Tamiya Twin-Motor Gearbox

This motor gearbox used on Tamiya track rover chassis with Arduino microcontroller which has motor controller, so no need to use an external motor controller. The positive terminals of the Motor 1 and Motor 2 connected to the M1+ and M2+ on microcontroller and the negative terminals of the Motor 1 and Motor 2 connected to the M1- and M2- on microcontroller respectively. Encoders are used to control the speed and direction of the motion.

3.4.2. Micro Metal Gear Motor

The micro metal gear motors are very tiny, made up of metal with gearbox and D shaped shaft as shown in figure 3.16. These motor are available in wide range of gear ratio, among which we are using 50:1 gear ratio. The dimension of the motor is given in figure 3.17 with the gearbox attached to it. These motors are easy to assemble with chassis and very powerful to run at high-speed.

Figure 3.16 - Micro Metal Gear Motor.

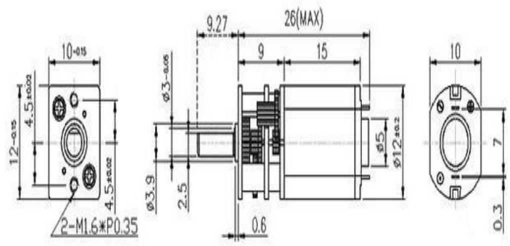


Figure 3.17 - Motor Dimensions with Gearbox

The technical specifications of the micro metal gear motor are given below,

- 13000 rpm @ No Load
- 50:1 Gear Ratio
- 260 rpm @ 6V
- 40mA @ 6V
- 360mA Stall Current @ 6V
- 10 oz inches Torque @ 6V

3.4.3. Solarbotics GM9 Gear Motor

The Solarbotics GM9 Gear Motor is very fast compared with the other version of the GM series as shown in figure 3.18. This brushed dc motors are preassembled with gearbox and enclosed. This motor is cheap, lightweight with high-speed ratio 143:1 with high torque. To control the speed and direction of the motors we are using motor controller shield with Arduino Uno microcontroller.

Figure 3.18 - Solarbotics GM9 Gear Motor (© Robotshop INC)

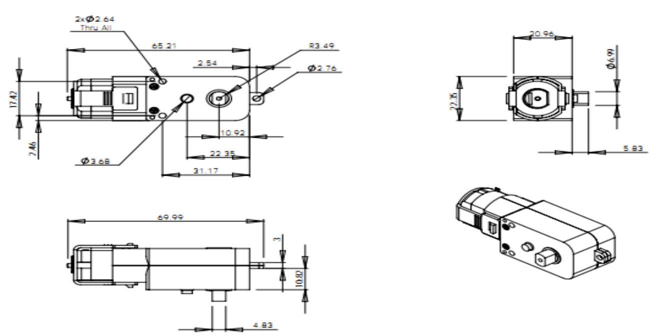


Figure 3.19 - Dimensions of the motor

The figure 3.19 shows the dimensions of the motor and following are the technical specification of the motor:

- Description: 90-degree
- Gear ratio: 143:1
- 3 V Operation:

  No-load RPM: ~ 40,

  No-load current: ~ 50 mA.

  Stall current: ~ 400 mA.

  Stall torque: ~ 44 oz-in
- 6 V Operation:

  No-load RPM: ~ 78,

No-load current: ~ 52 mA,

Stall current: ~ 700 mA,

Stall torque: ~ 76 oz-in

3.4.4. Hitec HS-422 Servo Motor

The Hitec HS 422 servo motors as shown in figure 3.20 are the most used in robots application because of their durability, high performance and torque. By using these motors we have build two and three DoF small robot arms with a gripper attachment. The color coded wires are easy to connect with microcontroller. The black wire is ground, the red wire is power and third wire is signal from microcontroller.

Figure 3.20 - Hitec HS-422 Servo Motor (© Robotshop INC)

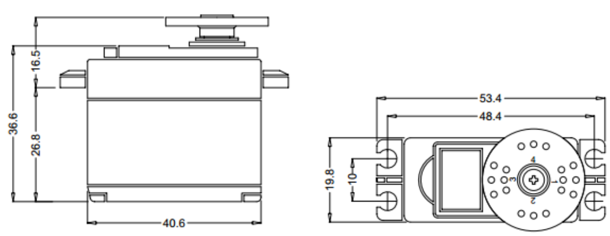


Figure 3.21 - Dimension of the HS 422 motor (© Arduino INC)

The figure 3.21 shows the dimension of the motor and following are the technical specification;

- Speed (sec/60o): 0.16
- Control Signal: Pulse Width Control
- Operating Voltage: 4 – 6 VDC
- Torque (Kg-cm/Oz-in): 4.1/57
- Size (mm): 41 x 20 x 37
- Weight (g/oz): 45.5/1.6

These motor are used in ultrasonic scanner kit, mounted on the top of the chassis which rotates and continuously scan for the obstacle or objects detection.

3.4.5. Motor Controller

We use the motor controller to drive the wheel motors as well as the microcontroller. The figure 3.22 shows the Pololu low voltage dual motor controller which is mounted on Rover 5 to control the speed and direction of the wheel motors. This low voltage dual motor controller is specially designed for the motors those required low voltage high current to drive. The left side motor positive terminal (Black wire) is connected to M0+ and negative terminal (Red wire) is connected to the M0- of the motor controller. The right side motor positive terminal is connected to the M1+ and negative terminal connected to the M1- on the motor controller. The Vcc terminal of motor controller is connected to the 5 V on microcontroller and the GND of the battery; motor controller and microcontroller are connected to each other. The SER pin of the motor controller is connected to the Pin 1 – Tx pin of the microcontroller and RST on motor controller connected to the pin on microcontroller. The complete wiring diagram for motor controller and microcontroller of the Rover 5 is shown in figure 3.23.

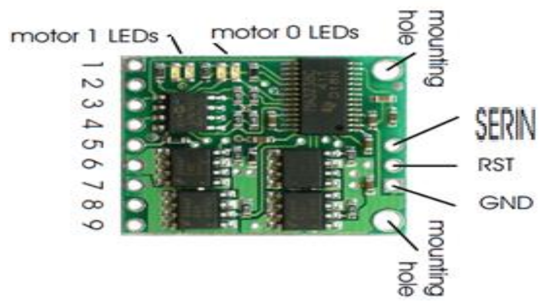


Figure 3.22 - Pololu Low Voltage Dual Motor Controller

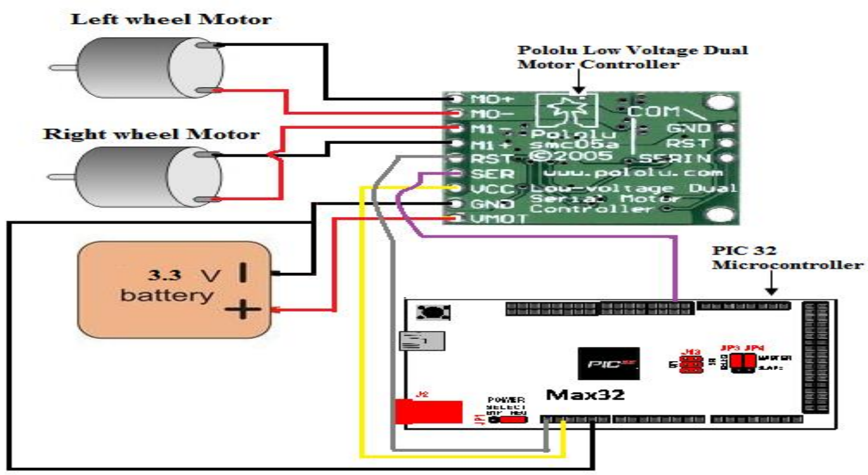


Figure 3.23 - Wiring Diagram for Motor Controller

For the other robots we are using Arduino compatible motor shield as shown in figure 3.24, which are easy to install on the Arduino microcontroller.

Figure 3.24 - Arduino Motor Controller (© Arduino INC)

The DFRobot Arduino Compatible Motor Shield (2A) uses L298P chip which allow driving the two 7-12V DC motors with maximum 2A current. This shield directly mounted onto standard Arduino Microcontroller and the speed control is achieved through conventional PWM which can be obtained from Arduino's PWM output Pin 5 and 6. The enable/disable function of the motor control is signaled by Arduino Digital Pin 4 and 7. The motor shield can be powered directly from Arduino or from external power source.

3.4.6. Small Manipulator Arm with Gripper

To add more flexibility and modularity to the robot swarms, small manipulator arms with gripper are attached on the chassis. These arms are with 2 or 3 Degree of Freedom (DoF) and were built in the UB RISC lab, using the off-the-shelf materials such as aluminum plates, plastic materials, nut, and screws. In theory, advanced modularity and versatility is easy to explain, but at the hardware level it’s difficult to achieve and implement. The figure 3.25 shows images of the small arm with gripper mounted on robot rovers and actuated using Hitec HS-422 Servo Motors. The gripper can grip and can rotate to grab objects or for connecting with other robots in the swarm. The jaws of the gripper can be open up to 1.3” and the wrist rotates 180 degrees approximately. The figure 3.25 shows images of robot arm installed on Rover’s.

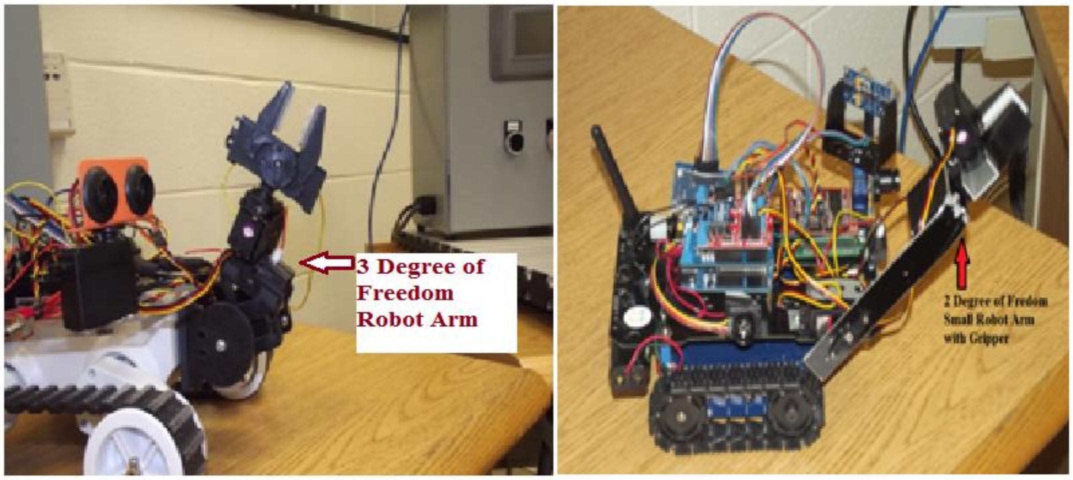


Figure 3.25 - Images of Robot Arm Mounted on Swarm

The mechanical design of robot manipulator arm involves the selection of motor gear ratio, material to be used for constructing the arm, gripper opening, number of degree of freedom. The design criteria also involve the torque and speed calculation depending on the application. Torque is a measure of how much force acting at the joint, which causes that joint to rotate. Mathematically, torque is cross product of the link distance and force.

$$T = F * L \quad \text{N-m}$$

Where, F = force acting on the motor

L = length of the link

The force (F) acts at a link length (L) from a joint. If we consider the vertical plane, the force acting on an object is the product of acceleration due to gravity and its mass, as given below,

$$F = m * g \text{ N}$$

Where, m = mass to be lifted by the motor

g = gravitational constant = 9.8 m/s

The force is also considered as the weight of an object, therefore, above equation can be re-write as,

$$W = m * g$$

The torque required to hold a mass at a given distance from a joint is,

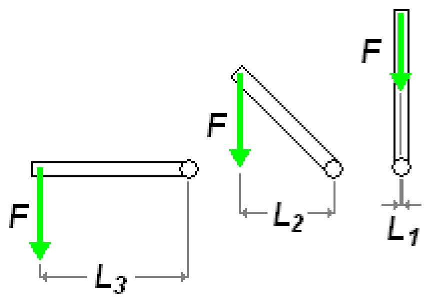
$$T = (m * g) * L$$

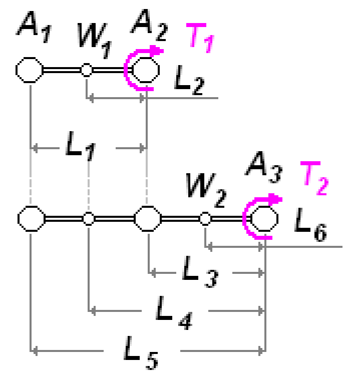
In order to estimate the torque required at each joint, we must choose the worst case scenario. The base joint carries the maximum load as it carries the weight of upper joint motors and links. The torque calculation must be done for each lifting actuator, by multiplying downward force and the linkage lengths as shown in figure. The weight of the object being held (A1), multiplied by the distance between its center of mass and the pivot gives the torque required at the pivot. The weight and center of mass of the link is located at the center of link, so the torque must be calculated by adding these weights.

$$T1 = L1 * A1 + \frac{1}{2} L1 * W1$$

The torque required at the second joint is calculated by adding link weights, distance between joints as given in below formula.

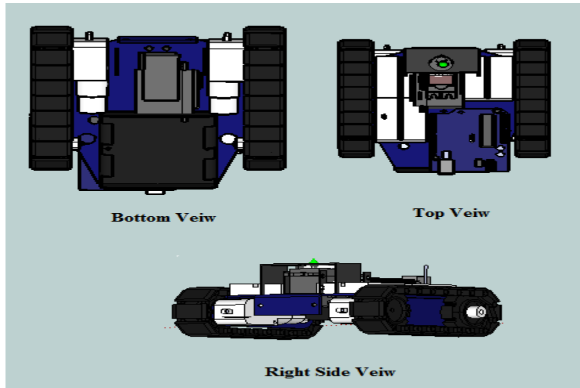
$$T2 = L5*A1 + L4*W1 + L3*A2 + L6*W2$$



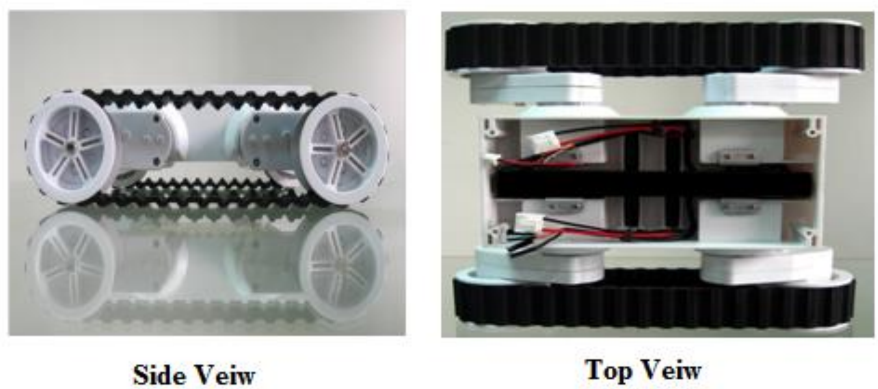


3.4.7. Chassis

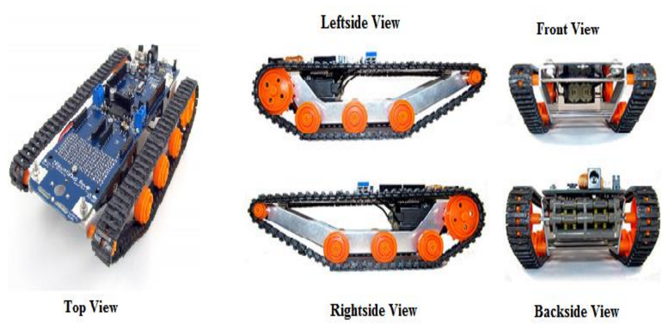
The body of robot is its chassis, on which all the sensors, wheels, actuators, microcontroller, camera etc are mounted. We are using the Dagu Rover 5, DFRobot Rover V2, Robot shop Rover, SRV1 chassis by keeping in mind about their size, shape and cost. We made some changes in these chassis according to our requirement so that we can add all the sensors, grippers and 2, 3 DOF small robot arms. The following figure 3.26 (A1, A2, A3, A4) shows the images of all robot chassis we are using before the modification and installation of the other parts of the robot swarm.



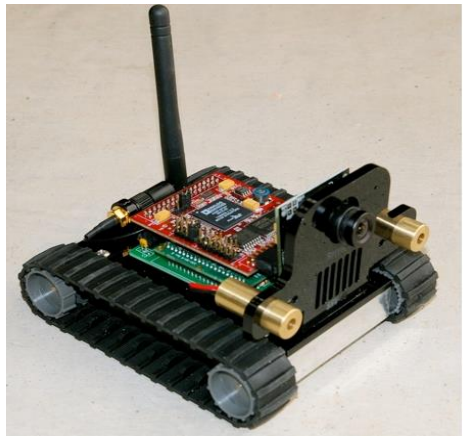
A 1 RobotShop Rover Chassis



A 2 Dagu Rover 5 2WD Chassis



A 3 DF Robot Rover V2 Chassis



A 4 SRV 1 Chassis

Figure 3.26 - Robot Development Chassis (A1, A2, A3, A4)

3.5 Communication and Control

3.5.1. Communication

One of the most important factors for more efficient cooperative robots is the communication among them and their environment [**68**]. Deploying a team of robot swarms to perform the specific task such as mapping, surveillance, wall painting, and rescuing, etc requires continuous communication between the robots swarm. In our previous survey paper [**69**], we describe all the methods of communication between the robots. Communication works in different way and it depend on the factors including communication range, environment, size of the swarm system, type of information to be sent and receive. In [**70**], the comparison between two well-known communication types – implicit and explicit has been made. The proposed robot swarm is decentralized in nature and they can communicate with each other, or/and host computer using a wireless network. Due to the advances in technology and microchip fabrication, electronic devices become more compact, smaller and consume low power. Nowadays, there are many hardware devices present in the market to accomplish the wireless communication for robot swarms. For communication, each robot swarm equipped with X-Bee module or Bluetooth Bee module or PmodWiFi module. X-Bee series 1, Bluetooth Bee and PmodWiFi are compatible to each other and use same protocol for communication. The X-Bee and Bluetooth Bee use the serial transfer mode (Tx and Rx) while the PmodWiFi uses SPI mode for transmitting and receiving the data. Using these modules we have created an ad hoc communication network.

3.5.1.1. PmodWiFi Module

The PmodWiFi expansion module is an interface board which can add Wi-Fi communication to our swarm system. The figure 3.27 shows the PmodWiFi module and its pin description.

a) PmodWiFi module

Connector J1 – SPI Communications		
Pin	**Signal**	**Description**
1	~SS	Slave Select
2	MOSI	Master out/Slave in Data
3	MISO	Master in/Slave out Data
4	SCK	Serial Clock
5	GND	Power Supply Ground
6	VCC	Power Supply (3.3V)
7	~INT	Interrupt Output
8	~RST	Hardware Reset
9	~WP	Write Protect
10	HIB	Hibernate
11	GND	Power Supply Ground
12	VCC	Power Supply (3.3V)

b) Pin Discription

Figure 3.27 - PmodWiFi Module

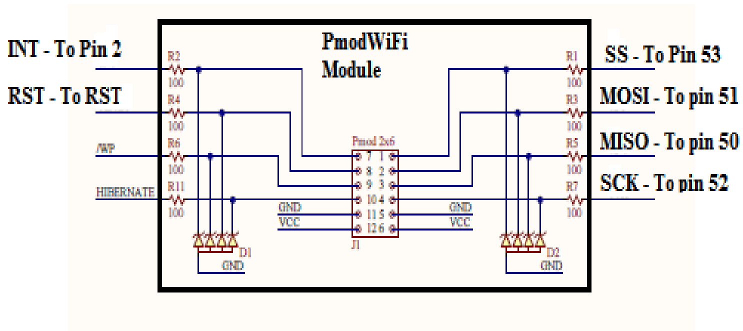


Figure 3.28 - Pin Diagram Connection

The PmodWiFi module uses SPI bus as a primary interface for communicating with PIC-Max32 microcontroller on Rover 5. The SPI bus uses four signals – SS, MOSI, MISO and SCK which corresponds to the signal selection, data in/ out and clock signal. The INT provides information of data availability and data transfer complete or not to the microcontroller respectively. The SS, MOSI, MISO, SCK, INT, RST pins of PmodWiFi module are connected to the pin number 53, 51, 50, 52, 2, RST of the microcontroller respectively as shown in figure 3.28.

3.5.1.2. XBee

The XBee is most popular and used communication transceiver module which allows making point-to-point or multipoint communication between robot swarms or computer. There are different modules of XBee available in the market; among those we chose series 1 of XBee for our robot swarm. This module supports the ZigBee networking protocol and unique, low cost, consumes less power for wireless networks. It uses Direct Sequence Spread Spectrum and each channel has over 65,000 unique network addresses available for point-to-point, point to multipoint, peer to peer communication.

Figure 3. 29 - XBee Series 1 (© Robotshop INC)

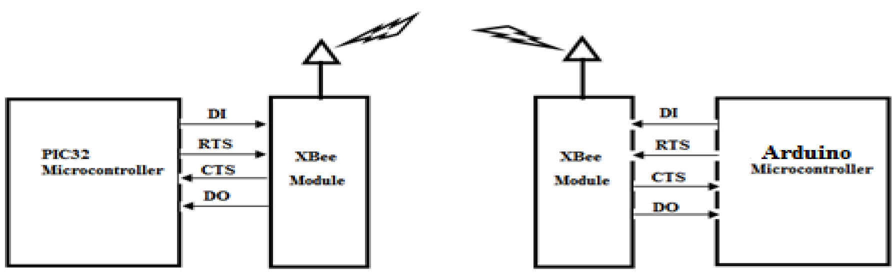


Figure 3.30 - Communication Between two microcontroller using XBee

The figure 3.29 shows the XBee module and figure 3.30 shows the communication set up between two different microcontrollers using transmit and receive pin over the wireless network using XBee. The ad hoc network is created using the XBee for communication between robot swarm. Following are the technical specification of the XBee series 1.

- 3.3V @ 50mA
- 250kbps Max data rate
- 1mW output (+0dBm)
- 300ft (100m) range
- Wire antenna
- 6 10-bit ADC input pins

- 8 digital IO pins
- 128-bit encryption
- AT or API command set

3.5.2. Controller

Controlling the robot is a really difficult task, especially for a swarm system. The robots in a multiagent system are controlled using centralized or decentralized methods. The drawbacks of centralized control are explained in our paper, so we decided to use centralized and/or decentralized control method depending on the task requirement. If the decentralized technique is applied, the hardware structure of robots should be highly redundant, but exploit simple and more robust control strategies. The brain for the robot is its microcontroller in which the user defined inference rules and knowledge base is stored. The performance of the robot depends on its microcontroller. The primary function of the controller is to route and manipulate the communications between other subsystems on the robot such as sensing platform, actuators, navigation system, and localization system. Robot swarms try to move the robots by sending the control signals to drive the motors. We use PIC32 and Arduino Uno microcontroller for our robot swarm. The programming language used for these controllers is C++ and both controllers are compatible with each other. Most of the parts used on this swarm team are bought from.

PIC32

Figure 3.31 shows the PIC controller which is very powerful controller, features a 32-bit MIPS processor core running at 80 MHz, 512K of flash program memory and 128K of SRAM data memory. In addition, the processor provides a USB 2 OTG controller, 10/100 Ethernet MAC and dual CAN controllers that can be accessed via add-on I/O shields. This controller mounted on Rover 5 and has 83 I/O which are as given below,

- 16 or 32-bit Timers
- 16 or 32-bit PWM
- 16 ch. 1 Msps 10-bit ADC
- 2x Comparators
- 5 x I2C™
- 4x SPI
- 6 X UART (with IrDA® encoder and decoder)

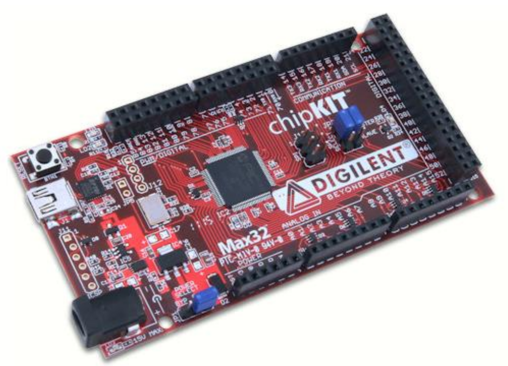


Figure 3.31 - PIC Max32 Microcontroller (© Diligent INC)

Arduino Uno

Arduino Uno is open source hardware platforms, which add flexibility in our robot swarms. This board based on the ATmega328, has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analogue inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. Ultrasonic sensors as well as sharp IR sensor are connected to the analogue input pins, encoders connected to the digital input pins of the controller. This board can be powered by USB port or by 3-6VDC an external power supply. Pin 0 and Pin 1 are used for TTL serial data receiver (Rx) and data transfer (Tx). The figure 3.32 shows an image of Arduino Uno microcontroller.

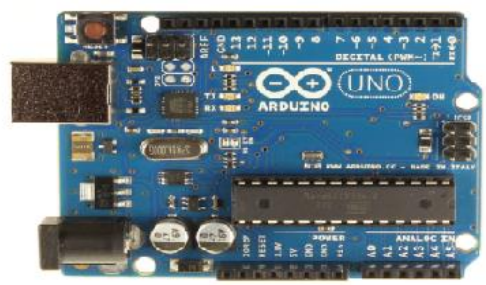


Figure 3.32 - Arduino Uno Microcontroller (© Arduino INC)

3.6 Power Supply

To keep the robot swarm running, we need to provide relatively long lasting power to them. We have chosen rechargeable Nickel Metal Hydride (NiMH) and Lithium Polymer batteries as a power supply for our robot swarms. These batteries are small in size, lighter in weight and easy to install on the chassis.

Nickel Metal Hydride battery has a high electrolyte conductivity rate which allows for high-power applications, and is cheaper than Li-Ion batteries, with high shelf life but self-discharge rate is higher than other batteries. On the other hand Lithium polymer batteries are another form of rechargeable batteries (LiPo) composed of several identical cells in parallel addition which increases discharge current. These batteries are expensive, slim, lighter in weight, and have stable overcharge. The current/power consumption for each robot in UB swarm is calculated and power management/saving technique is discussed in section.

CHAPTER 4: HARDWARE DESIGN AND IMPLEMENTATION

The main goal of this research work is to design and implement the heterogeneous mobile swarm robots. The proposed hardware architecture is inexpensive and can be used for the real life applications as well as for research purpose. This hardware platform is flexible, scalable and extensible for future development. The complexity of designing and physically implementing the heterogeneous robot swarm is greater when compared with the homogeneous robot swarms. There are several aspects involved in the development of robot swarm hardware, such as locomotion, actuation, navigation, size, appropriate sensors, cost, and communication. One of the challenges for robot swarm is its autonomy, as the robot must be aware of its battery life, self-localization etc.

First, we reviewed all the existing swarm hardware platforms and decided to use miniature, light-weight, low cost, easily modifiable and powerful platforms for building the swarm of robots. The Swarm robots developed so far are aimed to provide a research platform and not intended for real-world applications or vice versa. This modular hardware architecture consists of independent sensory units, actuator modules, and communication unit, that make swarm system scalable, modular and flexible such that more sensors and/or actuators can be added without modifying the overall architecture. The figure 4.1 shows an overview of the hardware design implementation.

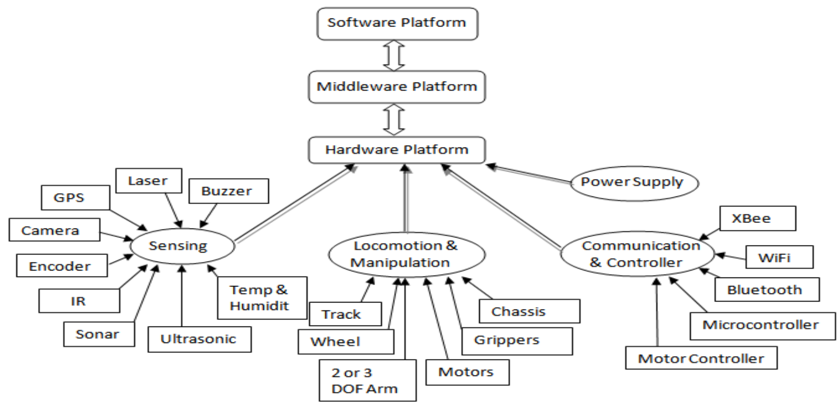


Figure 4.1 - An Overview of Hardware Architecture

There are multiple issues that have to be considered while designing and implementing the hardware platform for the heterogeneous robots. Following are the design goals for UB swarm of heterogeneous robots, such as:

- Each robot should be modifiable and compatible with a high performance microcontroller and should consume less power.
- Should provide modular and flexible platforms.
- They should be reconfigurable and provide easy support for the software as well as for the middleware.
- They should provide communication for indoor as well as outdoor applications
- They should have enough scope for future expansion of sensory units and actuators.
- The robot should be relatively of different size and shape with light weight, so that it can allow ease of movement and maneuverability.
- Each robot should be fully functional, autonomous and be able to continuously coordinate and communicate with other robots.

4.1 Power Consumption and Management

In swarm robotics, the cooperation among the individual autonomous robots depends on several design parameters such as communication, and management of resources. The power management and distribution in swarm robotics is of very high importance, which depends not only on the electronic design but also on its mechanical structure. To perform a task in an unknown environment, robots should be capable of great degree of autonomy and operate over a longer time. The autonomous mobile robots draw power from batteries carried on the chassis in order to provide the power to the onboard sensors, actuators, and communication modules. Batteries have a limited lifetime, due to which the operational time of the robots in the swarm is also limited. For successful completion of the tasks, the robot swarm must be continuously aware of the lifetime of its power source; therefore, management of power resources is necessary and vital for spending the available energy for robots swarm economically.

The overall power consumption can be calculated by adding the current consumed by each sensor, actuators, microcontroller and all other electronic components that are mounted on the robots. The selection of the battery depends on many factors such as size, power rating, capacity, power cycle, and cost. In the UB Swarm, we have five heterogeneous robots, and for each robot, we have to calculate how much power is consumed by robot. We also have to consider the other factors that affect the power consumption such as its working environment, type of terrain, elevation, how many times gripper close and pull an object. To power the UB Swarm, we have chosen Lithium Polymer batteries as a power source, which have several advantages such as high energy density, smaller size, and safe performance over the other types of batteries. In addition, these batteries have very low self discharge rate and retention capacity.

We measured the time for which sensors and actuators will be in use or active and multiply this time by their operating current, for example, if the ultrasonic sensor uses 20mA when on, and will be on 80% of the time, you get 0.8 x 20mA = 16mA.

Rover 1 -

Sr. No.	Component	Rating	Operating Time (%)	Current Consumption * No of Components	Total
1	Ultrasonic Sensors (SRF02)	4 mA	70%	2.8 mA * 2	5.6 mA
2	Ultrasonic Sensors (URM V2)	20 mA	100%	20 mA*1	20 mA
3	IR Sensors (Sharp)	33 mA	50%	16.5 mA * 1	16.5 mA
4	Temp and Humidity sensor	4 mA	10%	0.4 mA *1	0.4 mA
5	Servos (HS 422)	120 mA	50%	60 mA * 4	240 mA
6	Wheel Drive Motors	160 mA	100%	160 mA * 1	160 mA
7	Microcontroller (PIC)	90 mA	100%	90 mA * 1	90 mA
8	Encoders	4 mA	100%	4 mA * 2	8 mA
9	Motor Controller	10 mA	100 %	10 mA * 1	10 mA
10	Miscellaneous	100 mA	100 %	100 mA * 1	100 mA
				Total	650.5 mA

Table 4.1 - Total Power Consumption for Rover 1

On this rover, a 2000mAh Lithium-Polymer battery is used to supply the power, and the total power consumed by this rover is 650.5 mA. So the battery lifetime can be calculated as

Battery Life = Battery Capacity / Total power consumed or required for robot

= 2000mAh/650.5mA

= 3.07 Hrs.

Rover 2 –

Sr. No.	Component	Rating	Operating Time	Current Consumption * No of Components	Total
1	Ultrasonic Sensors (EZ1)	2 mA	70 %	1.4 mA * 4	5.6 mA
2	IR Sensors (Sharp)	33 mA	50%	16.5 mA * 1	16.5 mA
3	X - Bee	250 mA	80%	200 mA * 1	200 mA
4	Servos (HS 422)	120 mA	50%	60 mA * 2	120 mA
5	Wheel Drive Motors	250 mA	100%	250 mA * 1	250 mA
6	Microcontroller PCB (Arduino V3)	100 mA	100%	100 mA * 1	100 mA
7	Encoders	4 mA	100%	4 mA * 2	8 mA
8	Miscellaneous	150 mA	100%	150 mA * 1	150 mA
9	Ultrasonic Sensor (Seeedstudio)	15 mA	100%	15 mA * 1	15 mA
				Total	815.1 mA

Table 4.2 - Total Power Consumption for Rover 2

On this rover, a 2200mAh Lithium-Polymer battery is used to supply the power, and the total power consumed by robot = 815.1 mA. So the battery lifetime can be calculated as

Battery Life = Battery Capacity/Total power consumed or required for robot

= 2200mAh/815.1mA

= 2.69 Hrs.

Rover 3 –

Sr. No.	Component	Rating	Operating Time	Current Consumption * No of Components	Total
1	Ultrasonic Sensors(SRF2)	4 mA	70 %	2.8 mA * 2	5.6 mA
2	IR Sensors (Compound)	20 mA	50%	10 mA * 1	10 mA
3	Camera (Blackfin)	145 mA	80%	116 mA * 1	116 mA
4	Servos HS 422	120 mA	50%	60 mA * 3	180 mA
5	Wheel Drive Motors	73.7 mA	100%	73.3 mA * 2	146.6 mA
6	Microcontroller (Uno)	50 mA	100 %	50 mA * 1	50 mA
7	Ultrasonic Sensor (Ping)	20 mA	100%	20 mA * 1	20 mA
8	GPS/GPRS	100 mA	80%	36 mA * 2	72 mA
9	Laser Range Finder	40 mA	90 %	100 mA * 1	70 mA
10	Miscellaneous	100 mA	100%	100 mA * 1	100 mA
				Total	770.2 mA

Table 4.3 - Total Power Consumption for Rover 3

On this rover, a 2400mAh Lithium-Polymer battery is used to supply the power, and the total power consumed by robot = 770.2 mA. So the battery lifetime can be calculated as

Battery Life = Battery Capacity/Total power consumed or required for robot

= 2400mAh/770.2mA

= 3.11 Hrs.

Rover 4 –

Sr. No.	Component	Rating	Operating Time	Current Consumption * No of Components	Total
1	Ultrasonic Sensor (MaxSonar)	3.1 mA	80%	2.48 mA * 2	4.96 mA
2	IR Sensors (Sharp)	33 mA	50%	16.5 mA * 1	16.5 mA
3	Camera (Blackfin)	145 mA	80%	116 mA * 1	116 mA
4	Servos (HS 422)	120 mA	70%	84 mA * 1	84 mA
5	Wheel Drive Motors	100 mA	100 %	100 mA * 2	200 mA
6	Microcontroller Uno	50 mA	100 %	50 mA * 1	50 mA
7	Encoder	20 mA	100%	20 mA * 2	40 mA
8	Laser Range Finder	40 mA	90%	36 mA * 2	72 mA
9	X-Bee	250 mA	80%	200 mA * 1	200 mA
10	Miscellaneous	100 mA	100 %	100 mA * 1	100 mA
				Total	883.46 mA

Table 4.4 - Total Power Consumption for Rover 4

On this rover, a 2000mAh Lithium-Polymer battery is used to supply the power, and the total power consumed by robot = 883.46 mA. So the battery lifetime can be calculated as

Battery Life = Battery Capacity/Total power consumed or required for robot

= 2000mAh/883.46mA

= 2.2 Hrs.

Rover 5 –

Sr. No.	Component	Rating	Operating Time	Current Consumption * No of Components	Total
1	Ultrasonic Sensors	4 mA	70 %	2.8 mA * 2	5.6 mA
2	IR Sensors (Sharp)	33 mA	50%	16.5 mA * 1	16.5 mA
3	Servos	120 mA	70%	84 mA * 1	84 mA
4	Wheel Drive Motors	100 mA	100 %	100 mA * 2	200 mA
5	Microcontroller Uno	50 mA	100 %	50 mA * 1	50 mA
6	Encoders	20 mA	100%	20 mA * 2	40 mA
7	X-Bee	250 mA	80%	200 mA * 1	200 mA
8	Miscellaneous	100 mA	100%	100 mA * 1	100 mA
				Total	696.1 Ma

Table 4.5 - Total Power Consumption for Rover 5

On this rover, a 2000mAh Lithium-Polymer battery is used to supply the power, and the total power consumed by robot = 696.1 mA. So the battery lifetime can be calculated as

Battery Life = Battery Capacity/Total power consumed or required for robot

= 2000mAh/696.1mA

= 2.87 Hrs.

From the calculated power as shown in tables 4.1, 4.2, 4.3, 4.4, and 4.5, each robot consumes between 650 mA to 900 mA, which ensures continuous operation for a minimum of at least three hours. For this experiment, we decided to take three different sets of measurements. The first sets of measurements were taken while the robot rover is with load and full motion. The full load means, all the sensors, actuators, communication units, and microprocessor are in 100% working mode. So in 100% working mode, the discharged rate of battery will be very fast and robot rover will perform a task for three hours only as shown in figure 4.2 with a blue line. In the second set of measurements, the robot rover is in full motion with no load. In this experiment, only drive motors and only one sensor are in on mode while other sensors, actuators were in off mode. The discharged rate of battery is slower than the first case as shown in figure 4.2 with a red line. The robot rover performs the task longer than in the first case. To save battery power, we decided to do power management on the robot rover by choosing which sensor and actuator should be on for task completion. So in the algorithm, we control the on and off action of sensors, actuators, and drive motors depending on the task. In this power management method, sensors, actuators, and other components will be on only when needed; otherwise, they will go in sleep mode so that we can save battery power. The experimental measurements were plotted on graph as shown in figure 4.2 with a black line. We can see from the graph that robot performs task longer than the first two sets of measurements and the battery discharge rate is very slow.

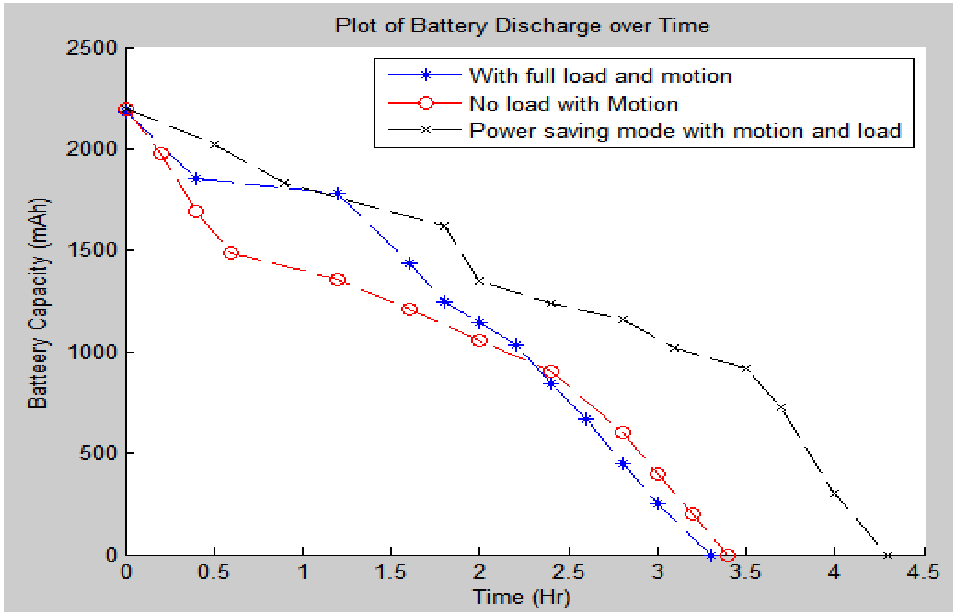


Figure 4.2 - Battery Capacity Vs Operating Time for Robot 1

For each robot of the UB swarm, current consumption is measured at different time intervals and plotted the graph in Matlab.

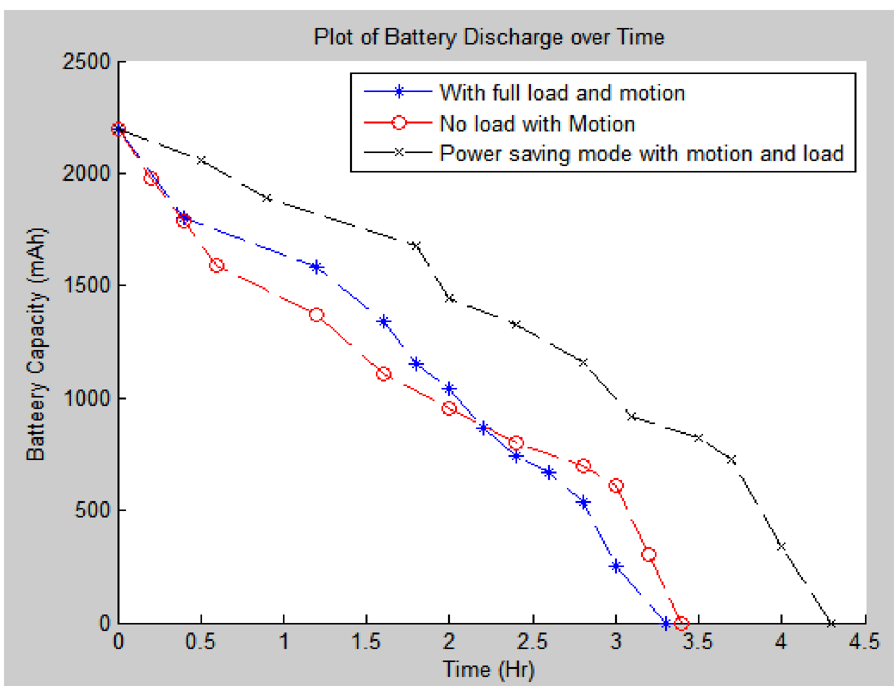


Figure 4.3 - Battery Capacity Vs Operating Time for Robot 2

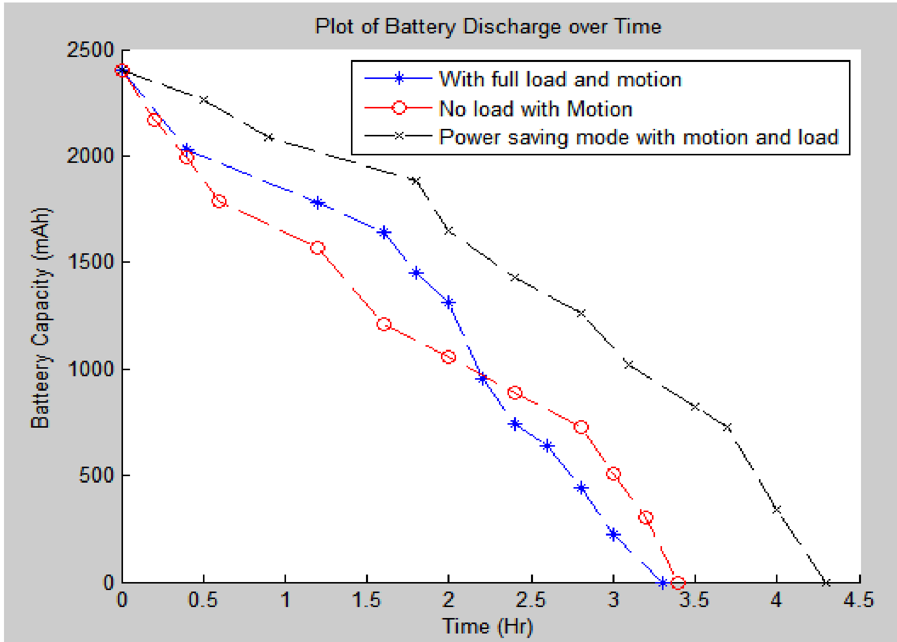


Figure 4.4 - Battery Capacity Vs Operating Time for Robot 3

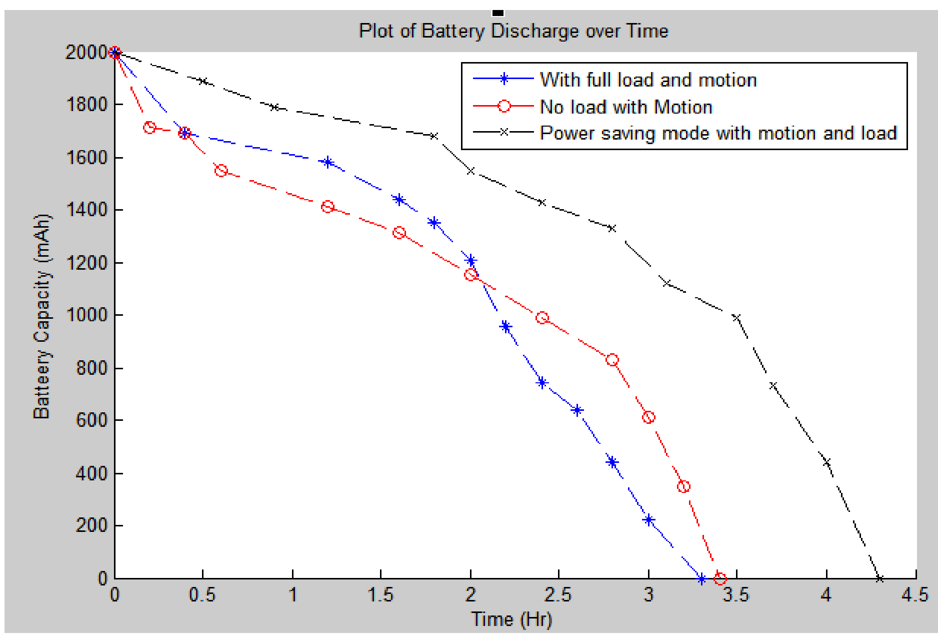


Figure 4.5 - Battery Capacity Vs Operating Time for Robot 4

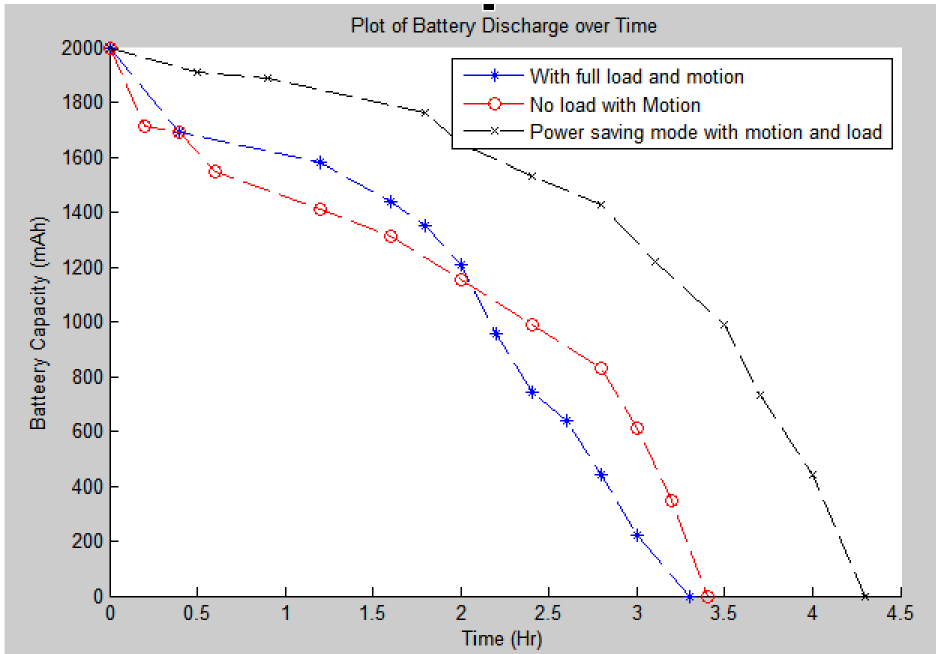


Figure 4.6 - Battery Capacity Vs Operating Time for Robot 5

The experimental measurement shows that the battery life is extended by 45 to 80 minutes by using power management technique.

4.2 Fault Detection

A fault is a sudden, unexpected change in behavior of the robot which hampers or disturbs the normal operation of the robot in the swarm. It is essential to detect the fault in the robot swarm before focusing on the fault tolerance. First we studied the types of fault that can occur in robots during a given task or in the working environment. The fault in robot swarm can occur at the physical level or at the software level. The physical level faults are related to hardware of robot such as damaged sensors, broken wheel, motors, short circuit in communication unit, while the software level faults are related with communication, algorithms as shown in figure 4.7.

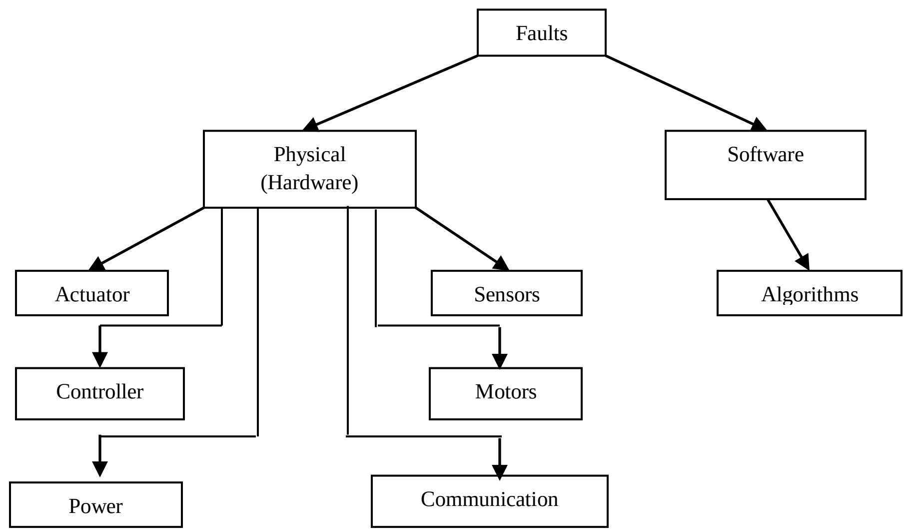


Figure 4.7 - Type of Faults

We can use the sensory data for fault detection that enables the robot to discover during normal operation and by using clustering technique to generate the probabilistic state diagram by putting boundary limits. We can also use the isolated software component to monitor the data flow, and if there is change in data flow, it will give a signal to the control program. We have assigned an ID for each robot, so if any fault occurs other robots in the swarm will know which robot has a fault. Following are the ID's assigned to each robot in UB swarm system:

Robot1 - UB1,

Robot2 – UB2,

Robot3 – UB3,

Robot4 – UB4,

Robot5 – UB5.

We can detect the fault in wheel or drive system by using encoder readings. If we do not read or get any feedback from the encoder, then there is a fault in the wheel or motor. Fault in other sensors can be determined by checking that the input pin on microcontroller is receiving any voltage or not. The faulty robot also sends a signal to the central system (operator), if it is in the centralized communication mode. The message signal contains the robot ID and the error code. If the other robot does not replied to robot within a certain time, there is a fault in communication unit. We have assigned tag for each fault such as given below:

F1: Sensor Failure
F2: Motor Failure
F3: Communication Failure
F4: Controller Failure
F5: Power Failure
F6: All System Failure

Whenever a fault occurred on any one of the robot of UB Swarm, that particular robot communicates to all the other robots about the fault and also central computer.

The Pseudo code for this fault detection for the micro-controller is given below,

```
1: if not timeout and ENQ received then
2:  send ACK to HostPC
3: else
4:  run robot
5: end if
6: while TRUE do
7:   wait for fault check
8:    if robot in fault then
9:      reply True
```

```
else
  reply False
end if
  check for fault
    if fault in sensor send F1 to HostPC AND other robot
    else if fault in motor send F2 to HostPC AND other robot
    else if fault in communication send F3 to HostPC AND other robot
    else if fault in controller send F4 to HostPC AND other robot
    else if fault in power send F5 to HostPC AND other robot
    else if fault in All system send F6 to HostPC AND other robot
  end if
end while
```

Fault tolerance is an ability of the swarm system to continue its operation in presence of a fault. The faulty robot or component not only affects the task completion process but also has affects on the other robots in the swarm. The fault tolerance can be achieved by hardware redundancy or software redundancy. In the hardware redundancy, we can use exactly the same type of hardware as a backup on the robot i.e. replication of the same hardware. This is a common approach for fault tolerance in sensory units. Having multiple sensory modules can act as a good fault tolerance measure. The redundant sensors can only be activated when a fault on the primary sensor is detected. If any fault occurs in any one of the sensors or components, the faulty sensor or component will be replaced by the secondary component or sensor. Adding the extra hardware will raise the other issues such as battery life, size and weight of the robot, cost. If a motor failure, controller failure, or communication failure is detected, in such case the faulty robot will be removed from the operation or task.

CHAPTER 5: RESULTS

We designed and built the UB swarm system composed of five heterogeneous robots. Each robot in UB swarm is fully autonomous, mobile, modular and capable of performing simple, basic tasks such as obstacle avoidance, manipulating objects, autonomous navigation, and perception of the environment. The hardware architecture is very flexible and we can connect any type of sensors without doing any major modification to it. The detail about sensors, actuators, controllers, communication unit, relative positioning system used on UB rovers is given in Table 5.1.

Sr No	UB Swarm	Sensors	Actuation	Controller	Communication	Chassis	Positioning System
1	Rover 1	Ultrasonic, IR, Temperature, Humidity, Encoder	Track wheeled, 3 DOF Robot Arm, Gripper	PIC32, Pololu Motor Controller	PmodWiFi,	Dagu Rover 5	Rangefinder,
2	Rover 2	Sonar, IR, Encoder, GPS	Track wheeled, Gripper, 2 DOF Robot Arm	Arduino V2	XBee	DF Robot Rover	IR, GPS
3	Rover 3	Laser, Ultrasonic, Camera, IR	Track Wheeled, Gripper	Arduino Uno, Motor Controller	XBee, Bluetooth Bee	Roboshop Rover	Laser, Camera, IR
4	Rover 4	Laser, Sonar, Camera	Track	Arduino Uno, 1000mips 500MHz Analog Devices Blackfin BF537	XBee, Camera	SRV1	Camera, Laser Range Finder
5	Rover 5	Ultrasonic, IR,	Track	Arduino Uno, Motor Controller	XBee	SRV1	XBee, IR

Table 5.1 - Detail of UB Heterogeneous Swarm Robots

The hardware platform of the swarm robot designed and built at the University of Bridgeport is compared with previously developed hardware platforms by considering major factors such as, sensory system, autonomy, fault tolerance, communication, and manipulability.

Sr. No.	Factors	S-Bot	UB Robot Swarm
1	Sensory System	IR sensors, Camera, torque sensor, microphone, speaker	IR, ultrasonic and laser sensor Camera, Temperature and Humidity, Encoder, Buzzer.
2	Autonomy	2 to 3 hrs	4 to 5 hrs
3	Fault Detection	Yes – LED indication	Yes
4	Manipulability	Yes – Gripper	Yes – Gripper
5	Communication	IR sensors, Wifi	XBee, IR sensor, Bluetooth
6	Hardware	Homogeneous	Heterogeneous

Table 5.2 - Swarm Bot Vs UB Robot Swarm

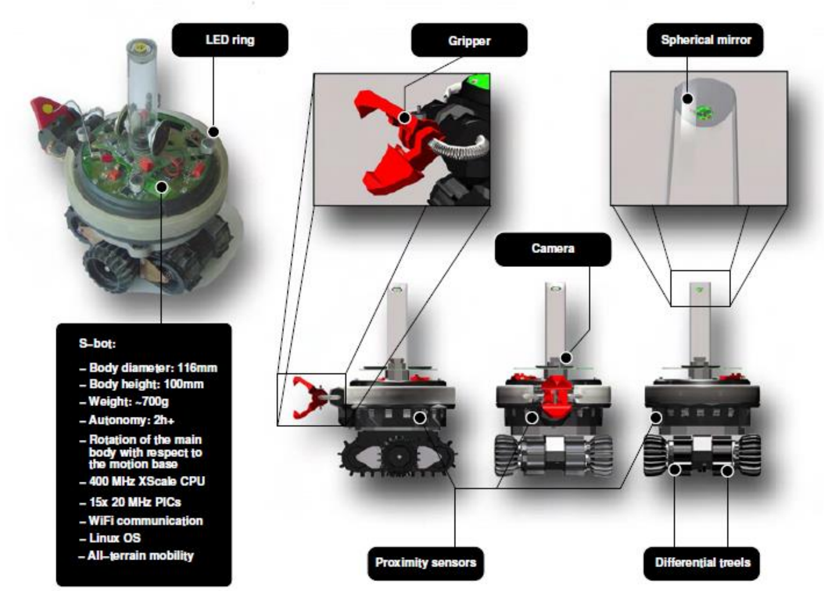


Figure 5.1 - Swarm Bot

The comparison between Swarm Bot and UB Swarm is given in table 5.1. The swarm bot is equipped with all the sensors such as a color omnidirectional camera, 16 lateral and 4 bottom infra-red proximity sensors, 24 light sensors, 3 axis accelerometer, two humidity sensors as well as incremental encoders and torque sensors. The Swarm Bot is also equipped with grippers for interconnection with other swarm bot. The power consumption is very high because of all these sensors. IR sensors cameras are the only two ways of communication with other bots and central control system, whenever the WIFI available.

Sr. No.	Factors	E Puck	UB Robot Swarm
1	Sensory System	IR, Camera, Accelerometer, Microphone, Speaker.	IR, ultrasonic and laser sensor Camera, Temperature and Humidity, Encoder, Buzzer, GPS/GPRS/GPM
2	Autonomy	2 Hrs	4 to 5 hrs
3	Fault Detection	No	Yes
4	Manipulability	No	Yes – Gripper
5	Communication	Bluetooth	XBee, IR sensor, Bluetooth
6	Hardware	Homogeneous	Heterogeneous

Table 5.3 - E Puck Vs UB Robot Swarm

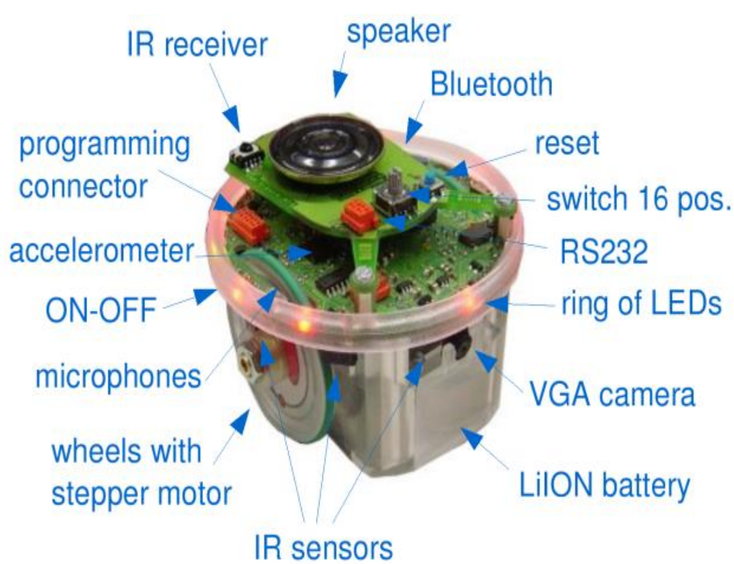


Figure 5.2 - E Puck

The E-puck is a homogeneous modular swarm robot, equipped with VGA camera, microphone, accelerometer, IR sensors, Bluetooth, speaker as shown in figure 5.2. The drawback of E-puck is that it uses a stepper motor for drive system and does not have any actuating arm for manipulation. The Bluetooth is the only way for communication, which has a limited range.

Sr. No.	Factors	Alice	UB Robot Swarm
1	Sensory System	IR Sensor,	IR, ultrasonic and laser sensor Camera, Temperature and Humidity, Encoder, Buzzer, GPS/GPRS/GPM
2	Autonomy	1.5 to 2 Hrs	4 to 5 hrs
3	Fault Detection	No	Yes
4	Manipulability	No	Yes – Gripper
5	Communication	IR	XBee, IR sensor, Bluetooth
6	Hardware	Homogeneous	Heterogeneous

Table 5.4 - Alice Vs UB Robot Swarm

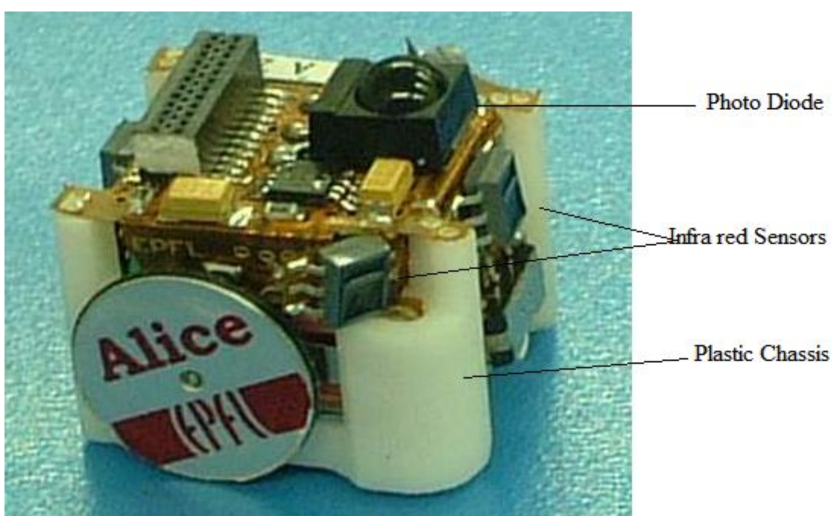


Figure 5.3 - Alice

Alice is a homogeneous robot equipped with IR sensors, two wheel drive system as shown in figure 5.3. Alice is very small in size, light weight and cheap enough for mass production. This robot platform can perform tasks only on soft surfaces and there should be no uneven surface because there is very little clearance between ground and robot chassis.

Sr. No.	Factors	Kobot	UB Robot Swarm
1	Sensory System	Compass, IR sensors, Omnidirectional Camera.	IR, ultrasonic and laser sensor Camera, Temperature and Humidity, Encoder, Buzzer, GPS/GPRS/GSM,
2	Autonomy	7 to 8 hrs	4 to 5 hrs
3	Fault Detection	No	Yes
4	Manipulability	No	Yes – Gripper
5	Communication	XBee	XBee, IR sensor, Bluetooth
6	Hardware	Homogeneous	Heterogeneous

Table 5.5 - Kobot Vs UB Robot Swarm

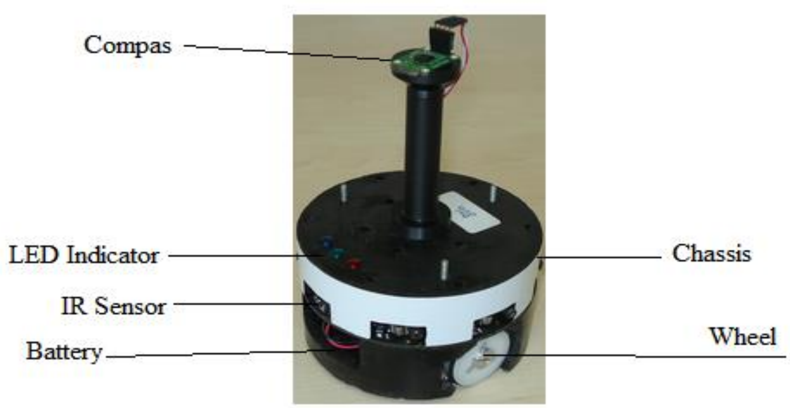


Figure 5.4 - Kobot

The Kobot robot, as shown in figure 5.4, is a homogeneous, modular, hybrid platform, equipped with IR sensors, compass, and an omnidirectional camera on the top with Zig-Bee module for wireless communication using IEEE802.15.4 protocol. These swarm robots are not capable of fault detection and tolerance if any fault occurs during operation.

Sr. No.	Factors	Jasmine	UB Robot Swarm
1	Sensory System	IR, Color Sensor, Touch Sensor, Distance Sensor	IR, ultrasonic and laser sensor Camera, Temperature and Humidity, Encoder, Buzzer, GPS/GPRS/GPM
2	Autonomy	2 – 3 hrs	4 to 5 hrs
3	Fault Detection	No	Yes
4	Manipulability	No	Yes – Gripper
5	Communication	IR	XBee, IR sensor, Bluetooth
6	Hardware	Homogeneous	Heterogeneous

Table 5.6 - Jasmine Vs UB Robot Swarm

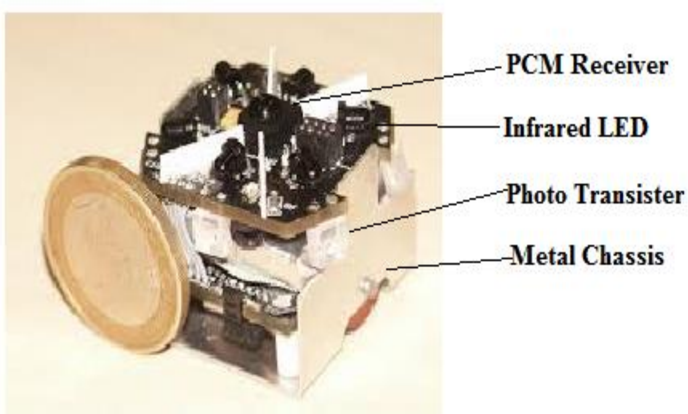


Figure 5.5 - Jasmine

The Jasmine micro robot platform is a reliable, cheap, and homogeneous swarm robot system, equipped with IR sensors, color sensors, and touch sensors as shown in figure 5.5. This robot platform has been only tested in laboratory, not in real field and also can be used only for indoor application. The wheels used for locomotion are small and thin, which makes them very difficult to move over the obstacles or on uneven surface.

On the other hand, UB robot swarm is heterogeneous and equipped with whatever necessary sensors and actuators required for all tasks. The UB robot swarm has plug and play approach due to which we can simply detach the unnecessary sensors, actuators and save current consumption on each robot which increases the battery life. Also UB robot can communicate with other robots in UB swarm by using WiFi, IR, Bluetooth, and XBee. The UB robot swarm equipped with IR, Ultrasonic, Sonar, Laser range finder, Humidity, Temperature, Buzzer, GPS/GPRS/GSM, Camera, LED, Bluetooth, Pmod WiFi, XBee and two or three degree of freedom small manipulators. These robots can communicate with each other as well as control system by centralize or decentralize communication.

The detail of each robot in UB swarm is given below:

A. Rover 1

The chassis used for this rover is the Dagu rover 5 with two wheel motor control. This chassis is slightly modified according to our need. For the object detection and obstacle avoidance, we use ultrasonic sensor (MaxSonar-EZ1 MB1010 Sensor) for long distance and infrared sensor (Sharp IR Sensor) for short distance measurement, depending on the task requirements they can be activated. The ping ultrasonic sensor mounted on the servo motor, so that we can scan continuously for obstacle avoidance. We also have a temperature and humidity sensor for sensing the environment, whether it is suitable for robot to operate or not. Encoders are used for determining the location or position of the robot by odometric principle. The three degree of freedom manipulator arm is attached to the chassis with gripper. The small arm is lighter in weight and built at our lab using plastic material. This arm can be used for pulling, grabbing, lifting or connecting to its neighbor robot, depending on the situation. The wheel motors are controlled by external motor controller (pololu). The microcontroller used on this rover is PIC32 with 84 I/O, features a 32-bit MIPS processor core running at 80 MHz for communication, Pmod WiFi module is used. The figure 5.6 shows the images of Rover 1,

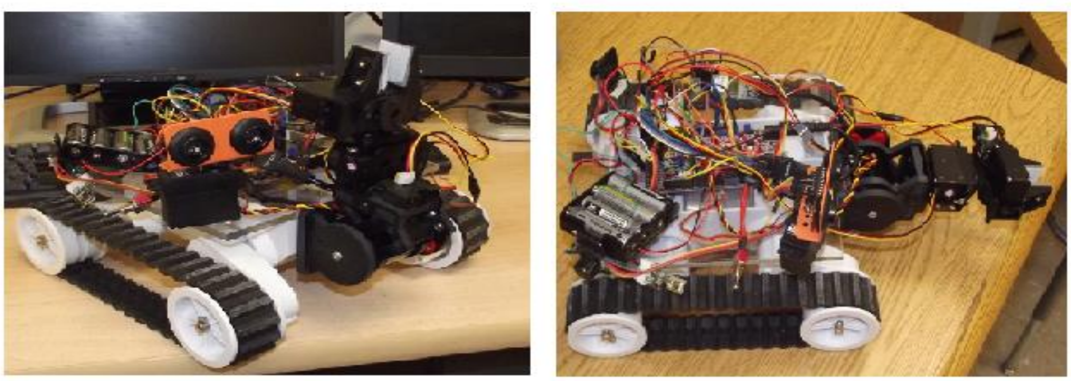


Figure 5.6 - Images of Rover 1

B. Rover 2

All the hardware components are mounted on the DF robot rover chassis. We made changes on this chassis to save battery power and for better performance. The Dual Tamiya gear motor with encoder is used for the locomotion. On this rover, we have sonar sensor (MaxSonar-EZ1 MB1010 Sensor) mounted on rear corners, and left and right side. The microcontroller Arduino V2 can communicate with other robots using XBee or Bluetooth Bee. Sharp IR sensor also used for detecting short distance measurement. We also use continuous scanning ultrasonic sensor with HS422 servo motor. Gripper with 1 degree of rotation is attached to the front of the chassis. The figure 5.7 shows an image of UB rover 2.

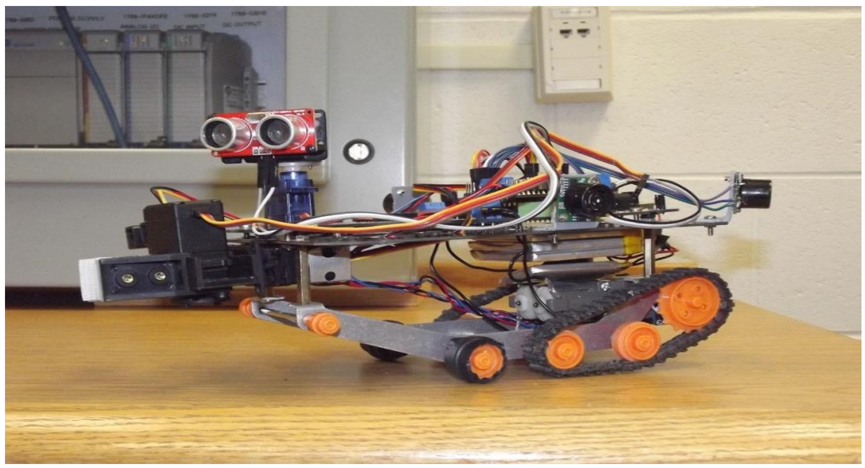


Figure 5.7 - Image of UB rover 2

C. Rover 3

The figure 5.8 shows an images of UB rover 3, on which compound infrared sensor mounted on the one side while ping sensor with continuous servo motor scanner kit on the front. We also use Blackfin camera with XBee communication module. A small two degree of freedom manipulator arm is attached on the side. The actuation of the manipulator is controlled by HS 422 servo motor.

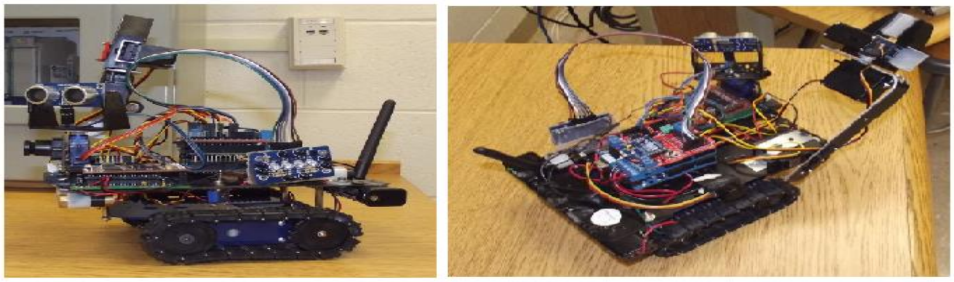


Figure 5.8 - Images of UB rover 3

D. Rover 4

The SRV 1 chassis is used for rover 4, as shown in figure 5.9. This chassis uses two micro metal gear motor for locomotion. XBee module can be used for communication between host computers or between four different robots within 100 feet for indoor application or 300 feet for outdoor application. Ultrasonic sensor and laser range detectors are used for obstacle avoidance and navigation. The Blackfin camera module is mounted on the front side of the rover. Arduino Uno microcontroller is used as controller as well as motor controller is also used for wheel motors control. A small gripper is attached on the front for connecting to the other robots to form chain or pulling the objects.

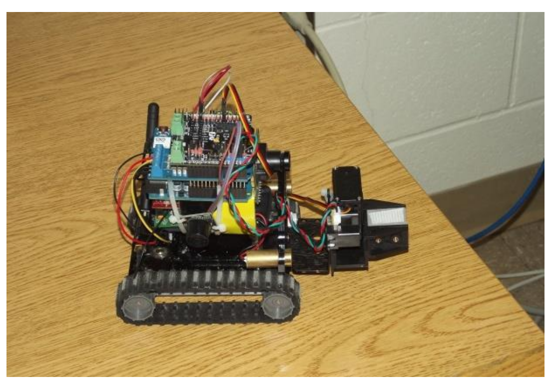


Figure 5.9 - Image of Rover 4

E. Rover 5

The rover 5 is use SRV 1 chassis with two micro metal gear motor. Infrared and ultrasonic sensors are used for distance measurement and obstacle avoidance. The XBee module is used for communication, to send/receive the data. The figure 5.10 shows an image of the rover 5. All the components operate on 5 VDC power supplies, Li-Po battery is mounted inside the chassis.

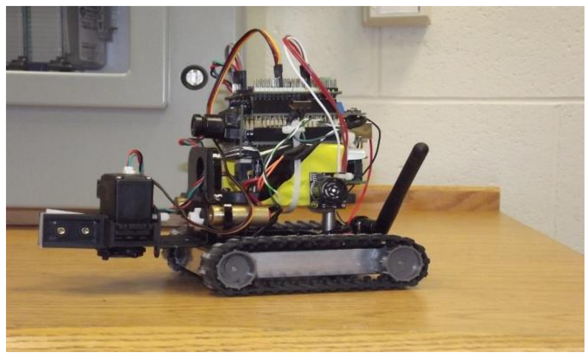
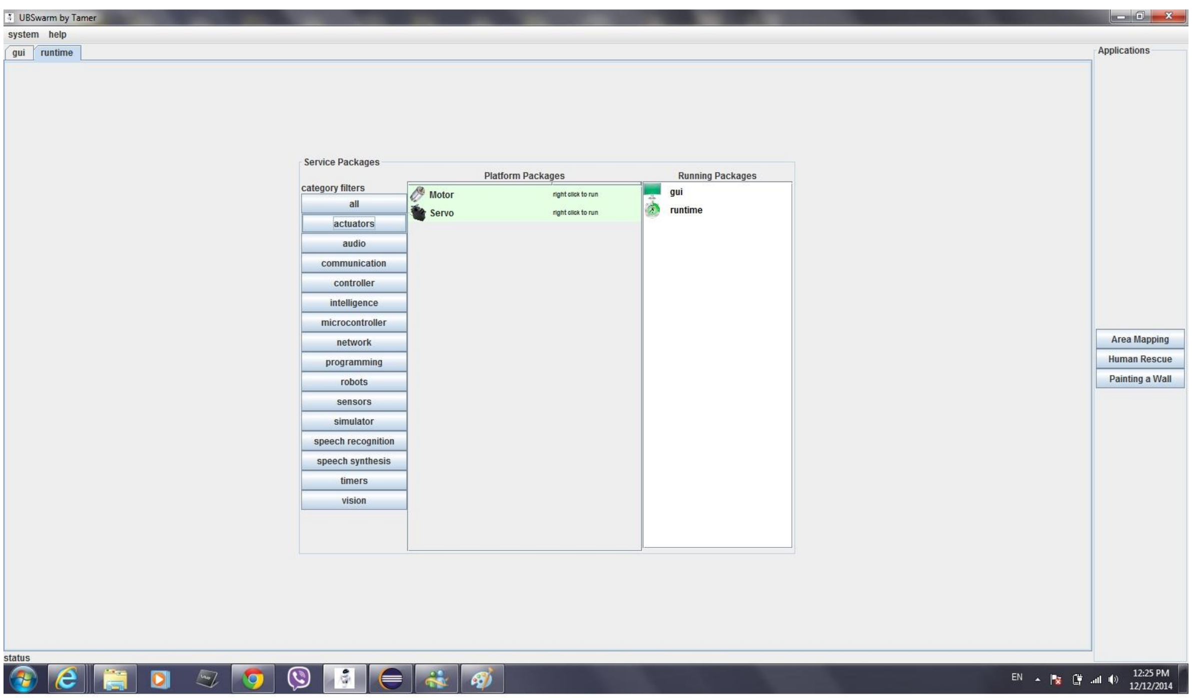
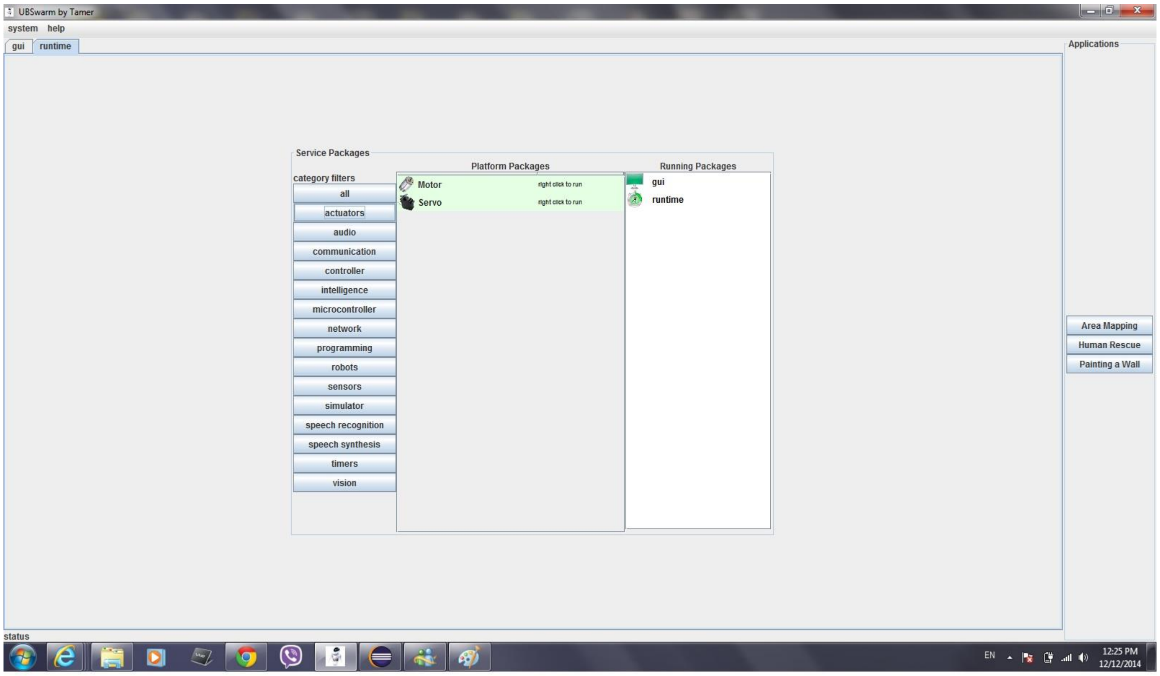


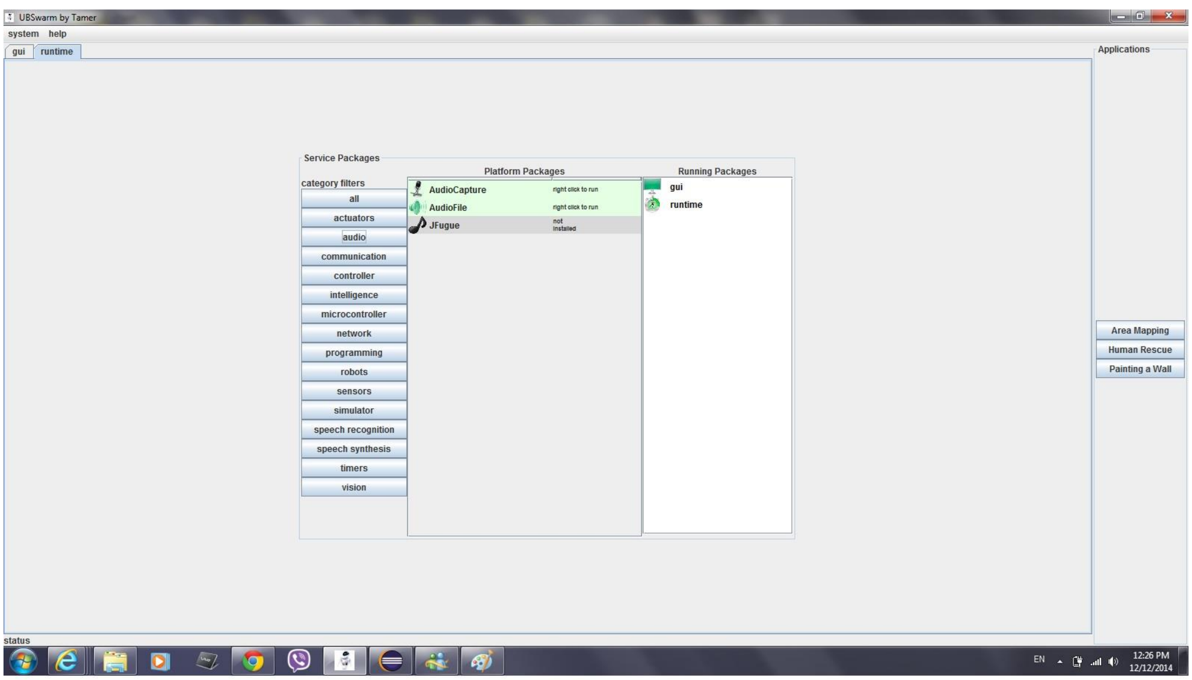
Figure 5.10 - Image of Rover 5

The hardware platform designed and built for the UB robot swarm can be simply customized using the plug and play approach, in which an end user can choose the sensory units, actuators according to the task. When the hardware devices are plugged into the framework, they are automatically detected by the middleware layer, which loads the appropriate software and avails the device for applications usage. This automatic detection and configuration of devices makes it efficient and seamless for end users to add and use new devices and software applications. To simplify the hardware platform for end user, we have labeled and tagged all the wires and terminals of all sensors, actuators, and microcontroller so that end user can easily plug them without any expertise in robotics.

The plug and play approach of the UB Swarm makes the end user choose sensors, actuators, and any of the UB robots depending on the task requirement without doing major modification on the hardware platform. All the hardware parts are compatible with each other, so the end user can use any sensors or actuators with any one of the UB swarm robots as shown in figure 5.11. For example, user can use UB1 rover with three ultrasonic sensors, one IR sensor and camera and UB2 rover with two IR sensors, laser range finder for rescue task. The user can also use UB1 with IR sensor and two DOF actuator, UB2 with Camera, two ultrasonic sensors for painting task. The end user has to just turn off the power supply and then unplug the unused sensors and actuators from respective UB robots and plug the required sensors and actuators.




UBSwarm by Tamer
system help
gui runtime
Applications
Service Packages
Platform Packages
Running Packages
category filters
all
actuators
audio
communication
controller
intelligence
microcontroller
network
programming
robots
sensors
simulator
speech recognition
speech synthesis
timers
vision
Motor
Servo
gui
runtime
Area Mapping
Human Rescue
Painting a Wall
status


UBSwarm by Tamer
system help
gui runtime
Applications
Service Packages
Platform Packages
Running Packages
category filters
all
actuators
audio
communication
controller
intelligence
microcontroller
network
programming
robots
sensors
simulator
speech recognition
speech synthesis
timers
vision
AudioCapture
AudioFile
JFugue
gui
runtime
Area Mapping
Human Rescue
Painting a Wall
status

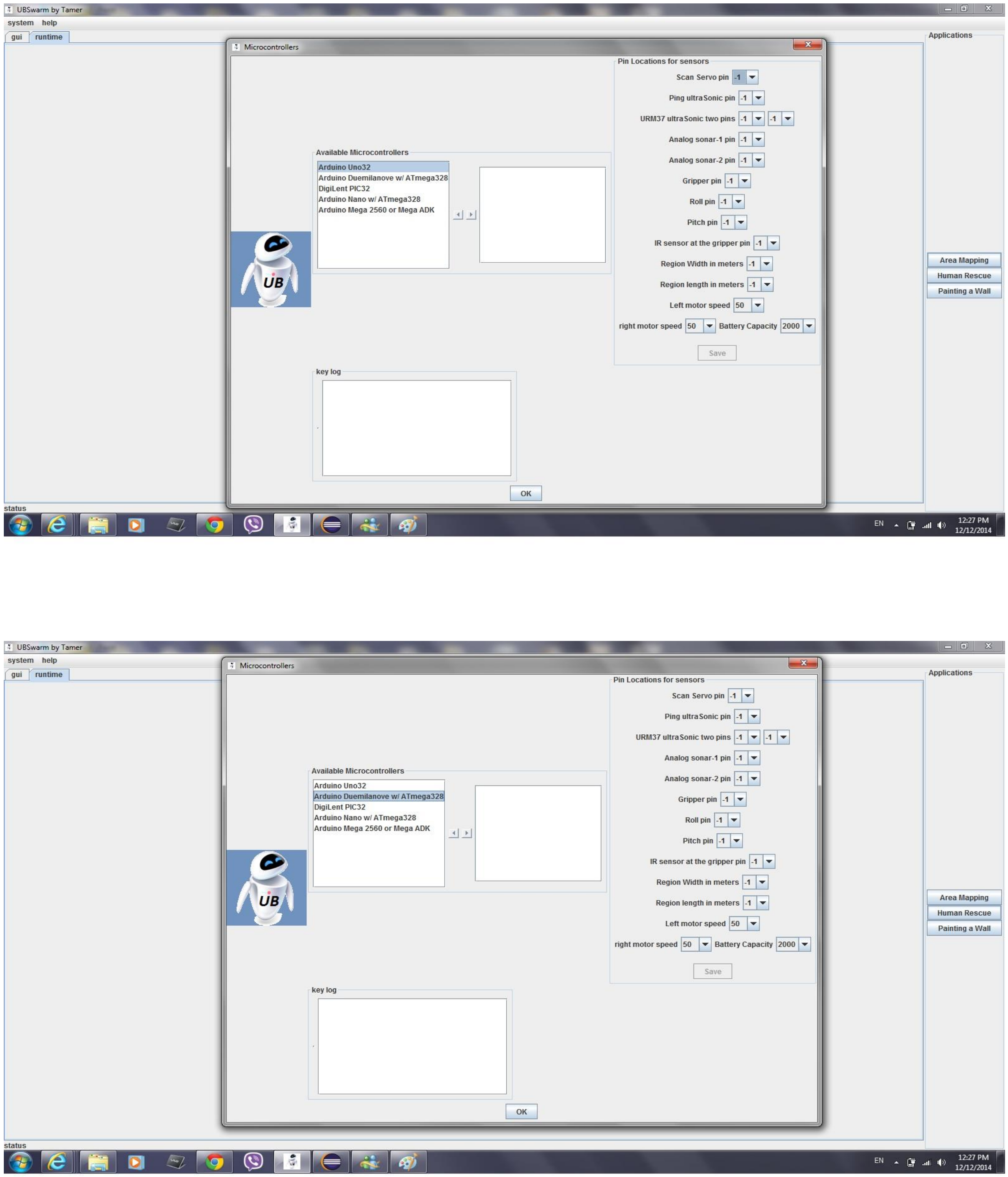


Figure 5.11 - UB Swarm Software Screen's.

This UB robot swarm was tested for a set of different experiments such as obstacle avoidance, object transportation, human rescue, wall painting and mapping. All the tasks are explained in this section and compare the results with increasing the number of robot in the swarm. To conduct this experiment we built small arena and initially robots placed randomly in the arena. A small web camera is mounted on the top of arena so that we can record the experiments.

5.1 Obstacle Avoidance

For swarm robots, avoiding an obstacle within dynamic and/or unknown environment is the fundamental issue. There are so many techniques are pointed in literature survey for obstacle detection and to avoid the collision. The UB robot swarm is fully capable of avoiding the collision and detecting the stationary as well as moving objects. The algorithm is implemented using C++ language and uploaded into the microcontroller. The sensors used for this application are IR sensor, ultrasonic sensor and camera. This sensor fusion data transmitted to the controller and this information also shared between the other robot swarm over the wireless network. In this test, robots start navigating from initial position towards the goal object. When the robot senses an obstacle in its path, it tries to stochastically overcome it and reaches to the targeted goal position.

5.2 Object Transportation

In robot swarm system, object transportation has sub functions such as pushing, grasping and caging. The object transportation using heterogeneous robots can have significant economical impact on industrial application such as packing, sorting etc. If this task is beyond the single robot swarm, then robot will send signal to the other robot for help to complete the task. The robots are equipped with 2 degree of freedom or 3 degree of freedom small manipulator arm with gripper which can be used for to grab the object. The camera or sensors are used to detect the goal object and then transport or lift the object against the external forces.

5.3 Human Rescue

Unstructured or unstable environment generated due to major accidents, natural disasters, and catastrophic events requires urgent intervention for rescuing humans. In such situations, the common operations are search, monitoring, rescue and transport. One of the tasks we tested using our robot swarm is to rescue a human. Our demonstrated example of search and rescue task shows the different integrated abilities of these heterogeneous robot swarm such as search, object detection, path planning and navigation, reconfigurability and rescue operation.

We describe a human rescue task and compare the results with increasing the number of robot in the swarm. We created a dummy human lying on ground inside the arena and robot swarm tries to rescue that dummy human by pulling it to a safe location. Initially we deploy only two robots of UB swarm for this task and record the time required by them to finish the task. After that we add one more robot to do the same task and recorded the time required for them to complete it. We did the same experimental task by deploying four and five robots of UB swarm and then compare the time required by them to complete the task. The result of these experiment shows that the time required for five robots is very less and execution is more efficient than in the other scenarios. The figure 5.12 and 5.13 shows human being rescued by using two and four robots of UB swarm respectively.

Figure 5.12 - Human rescue using two UB robot swarm

Figure 5.13 - Human rescue using four UB robot swarm

Table 5.7 shows the result of the human rescue task using UB robot swarm.

No of Robots	Time required (Minute)	Distance travelled (feet)	Task accuracy (%)
2	20	89	48
3	17	129	54
4	14	176	63
5	10	210	72

Table 5.7 - Experimental result comparison

5.4 **Catalog for Parts**

All the sensors, actuators, components, microcontroller used for UB robot swarm with their manufacturer, product code and distributor is given table 5.8 below,

Sr No	Components	Product Code	Manufacturer	Distributers
1	Ultrasonic Sensor SRF02	RB-Dev-20	Devantech Ltd.	Robotshop Inc.
2	Seeedstudio Ultrasonic Range Finder	RB-See-90	Seeed Technology Inc.	Robotshop Inc.
3	PING Ultrasonic Sensor	RB-Plx-73	Parallax Inc.	Robotshop Inc.
4	URM V3.2 Ultrasonic Sensor	RB-Dfr-11	DFRobot Inc.	Robotshop Inc.
5	LV-MaxSonar-EZ1 Sonar Module	RB-Max-01	MaxBotix Inc.	Robotshop Inc.
6	Sharp IR Range Finder	RB-Dem-02	Sharp Corporation	Robotshop Inc.
7	Dagu compound infrared sensor	RB-Dag-06	DAGU Hi-Tech Electronic Co.	Robotshop Inc.
8	Laser Range Finder	------	Surveyor Corporation	Alibaba.com
9	Humidity and Temperature Sensor SHT1x	RB-Dfr-68	DFRobot Inc.	Robotshop Inc.
10	Robot Chassis Encoder	RB-Dfr-218	DFRobot Inc.	Robotshop Inc.
11	Encoder Pair for Tamiya Twin Motor Gearbox	RB-Rbo-122	Robotshop Inc.	Robotshop Inc.
12	GPS/GPRS/GSM Shield V3.0	RB-Dfr-190	DFRobot Inc.	Robotshop Inc.
13	Blackfin Camera	--------	Surveyor Corporation	Alibaba.com
14	Bluetooth Module	RB-Dfr-10	DFRobot Inc.	Robotshop Inc.
15	Low Voltage Dual Serial Motor controller	RB-Pol-16	Pololu Corporation	Robotshop Inc.
16	Motor Controller Shield	RB-Dfr-58	DFRobot Inc.	Robotshop Inc.

17	Pmod WiFi Interface	RB-Dig-132	Diligent Inc.	Robotshop Inc.
18	XBee Series 1	RB-Spa-112	SparkFun Electronics	Robotshop Inc.
19	ChipKIT Max32 Arduino Compatible Microcontroller	RB-Dig-15	Diligent Inc.	Robotshop Inc.
20	Arduino Uno USB Microcontroller Rev 3	RB-Ard-34	Arduino	Robotshop Inc.
21	Buzzer Module	RB-Dfr-39	DFRobot Inc.	Robotshop Inc.
22	3.7V 2000mAh LiPo Battery	RB-Kow-10	Robotshop Inc.	Robotshop Inc.
23	7.4V 2200mAh LiPo Battery	RB-Dfr-160	DFRobot Inc.	Robotshop Inc.
24	7.4V 2000mAh LiPo Battery	RB-Kow-12	Robotshop Inc.	Robotshop Inc.
25	Tamiya Twin-Motor Gear Box	RB-Tam-01	Tamiya America Inc.	Robotshop Inc.
26	50:1 Micro Metal Gear motor	RB-Pol-60	Pololu Corporation	Robotshop Inc.
27	GM9 - Gear Motor	RB-Sbo-07	Solarbotics Ltd.	Robotshop Inc.
28	HS-422 Servo Motor	RB-Hit-27	Hitech Inc.	Robotshop Inc.
29	Little Grip Kit	RB-Rox-01	Lynxmotion Inc.	Robotshop Inc.
30	Rover V2 - Arduino Compatible Tracked Robot chassis	RB-Rbo-41	Robotshop Inc.	Robotshop Inc.
31	Robot Rover Chassis (Rubber Tracks)	RB-Rbo-118	Robotshop Inc.	Robotshop inc.
32	Rover 5 2WD Tracked Chassis	RB-Dag-38	DAGU Hi-Tech Electronic Co.	Robotshop Inc.
33	SRV 1 Surveyor Chassis	------	Surveyor Corporation	Alibaba.com
34	Wires	------	Robotshop Inc.	Robotshop Inc
35	Aluminum Plate	------	Home Depot	Home Depot

Table 5.8 - Part Catalog

CHAPTER 6: CONCLUSION

In this work we have outlined the drawbacks of the existing swarm hardware architectures. Most existing systems are homogeneous in nature composed of the same type robotic agents. Our survey outlines the limitation of having homogeneous swarm architecture. To overcome these limitations and add heterogeneous features to robotic swarms, we proposed novel heterogeneous hardware architecture called the UB Swarm.

UB swarm system consists of five robots which are heterogeneous in sensory units, microcontroller, functionality, and size. The proposed hardware architecture of heterogeneous robot swarm is designed, built and tested. We describe all the hardware components used to build UB robot swarm. We also present the results obtained from this work. The UB Swarm system uses both centralized and decentralized control strategies within the swarm. The robot-to-robot and robot-to-environment interaction provides the task oriented, simple collective swarm behavior.

CHAPTER 7: FUTURE WORK

Often a research project opens new dimensions and raises several more questions than it answers. This research work on hardware implementation of heterogeneous swarm robot still demands further investigation and development on some topics. Areas that can be developed further include:

1. More robots with different capabilities can be added to the UB Swarm, which can improve the system performance.
2. In the future, an increased effort in the research of new hardware approaches for fault detection and fault tolerance for heterogeneous swarm robots can be done. Robots need to be both safe and dependable before they can enter our homes and before they can be entrusted with critical mission and tasks, such as human rescue.
3. In the future, wireless modules can be added to the robot platform, which enable the robots to act at larger physical ranges.
4. New power management strategies can be implemented for power management such that the operating battery life can be increased.

REFERENCES

[1] A. J. C. Sharkey, "Robots, Insects and Swarm Intelligence," *Artificial Intelligence Review*, vol. 26, no. 4, pp. 255-268, Dec. 2006.

[2] M. G. Hinchey, R. Sterritt, and C. Rouff, "Swarms and Swarm Intelligence," *IEEE-Computer*, vol. 40, no. 4, pp. 111-113, Apr. 2007.

[3] M. Yim, et al., "Modular self-reconfigurable robot systems [grand challenges of robotics]," *Robotics & Automation Magazine, IEEE,* vol. 14, no. 1, pp. 43-52, Mar. 2007.

[4] R. Gross, E. Tuci, M. Dorigo, M. Bonani, and F. Mondada, "Object transport by modular robots that self-assemble," in *Proceedings 2006 IEEE International Conference on Robotics and Automation*, Orlando, FL, 2006, pp. 2558-2564.

[5] M. Yim, Y. Zhang, and D. Duff, "Modular robots," *Spectrum, IEEE*, vol. 39, no. 2, pp. 30-34, Feb. 2002.

[6] G. Bekey and J. Yuh, "Reviewing the issues of robotic self-X," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 4, pp. 6-7, Dec. 2007.

[7] J. v. Neumann, *Theory of Self-Reproducing Automata*, A. W. Burks, Ed. Urbana, IL, USA: University of Illinois Press, 1966.

[8] J. Suthakorn, "Toward Self-Replicating Robots: An Overview on Self-Replication Machine," *Journal of Thai Robotics Society*, vol. 3, no. 1, pp. 66-80, Apr. 2005.

[9] K. Lee and C. G.S., "Robotic self-replication," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 4, Dec. 2007.

[10] H. Bojinov, A. Casal, and T. Hogg, "Emergent structures in modular self-reconfigurable robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, San Francisco, CA, 2000, pp. 1734-1741.

[11] K. Kotay, M. Vona, and C. McGray, "The Self-reconfiguring Robotic Molecule," in *IEEE International Conference on Robotics and Automation*, Leuven, 1998, pp. 424-431.

[12] K. Stoy, W. Shen, and P. Will, "How to make a self-reconfigurable robot run," in *First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, 2002, pp. 813-820.

[13] F. Ducatelle, G. A. D. Caro, and L. M. Gambardella, "Cooperative Self-Organization in a Heterogeneous Swarm Robotic System," in *Genetic and Evolutionary Computation Conference* , New York, USA, 2010, pp. 87-94.

[14] M. Yim, et al., "Modular reconfigurable robots in space applications," *Journal of Autonomous Robots*, vol. 14, no. 2-3, pp. 225-237, Mar. 2003.

[15] K. Tomita, S. Murata, H. Kurokawa, E. Yoshida, and S. Kokaji, "Self-assembly and self-repair method for a distributed mechanical system," *Robotics and Automation*, vol. 15, no. 6, pp. 1035-1045, Dec. 1999.

[16] K. Støy, W. .-M. Shen, and P. Will, "On the use of sensors in self-reconfigurable robots," in *seventh international conference on simulation of adaptive behavior on From animals to animats*, MA, USA, 2002, pp. 48-57.

[17] J. Pollack, M. Bedau, P. Husbands, R. Watson, and T. Ikegami, "Designed and evolved blueprints for physical self-replicating machines," in *Ninth International Conference on the Simulation and Synthesis of Living Systems*, Cambridge, MA, 2004, pp. 15-20.

[18] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson, "Robotics:Self-reproducing machines," *Nature*, vol. 435, no. 7038, pp. 163-164, May 2005.

[19] M. W. Jørgensen, E. H. Østergaard, and H. H. Lund, "Modular ATRON: Modules for a self-reconfigurable robot," in *International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 2068-2073.

[20] D. Brandt, D. J. Christensen, and H. H. Lund, "ATRON robots: versatility from self-reconfigurable modules," in *International Conference on Mechatronics and Automation*, Harbin, China, 2007, pp. 26-32.

[21] S. Murata, E. Yoshida, H. Kurokawa, K. Tomita, and S. Kokaji, "Self-Repairing Mechanical Systems," *Autonomous Robots*, vol. 10, no. 1, pp. 7-21, Jan. 2001.

[22] C. Ünsal, H. Kiliççöte, and P. K. Khosla, "A modular self-reconfigurable bipartite robotic system: Implementation and motion planning," *Autonomous Robots* , vol. 10, no. 1, pp. 23-40, Jan. 2001.

[23] M. Yim, Y. Zhang, J. Lamping, and E. Mao, "Distributed Control for 3D

Metamorphosis," *Autonomous Robots* , vol. 10, no. 1, pp. 41-56, Jan. 2001.

[24] M. Yim, D. G. Duff, and K. D. Roufas, "PolyBot: a modular reconfigurable robot," in *International Conference on Robotics and Automation*, San Francisco, CA, 2000, pp. 511-520.

[25] A. Castano, W.-M. Shen, and P. Will, "CONRO: Towards deployable robots with inter-robots metamorphic capabilities," *Autonomous Robots*, vol. 8, no. 3, pp. 309-324, Jun. 2000.

[26] A. Castano, A. Behar, and Peter M. Will, "The CONRO modules for reconfigurable robots," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, pp. 403-409, Dec. 2002.

[27] M. Rubenstein, K. Payne, P. Will, and W.-M. Shen, "Docking Among Independent and Autonomous CONRO Self-Reconfigurable Robots," in *International Conference on Robotics and Automation*, Los Alamitos, CA, 2004, pp. 2877-2882.

[28] S. Murata, E. Yoshida, A. Kamimura, and H. Kurokawa, "M-TRAN: Self-Reconfigurable Modular," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, pp. 431-441, Dec. 2002.

[29] T. Fukuda and S. Nakagawa, "Dynamically reconfigurable robotic system," in *IEEE International Conference on Robotics and Automation*, Philadelphia, PA, 1988, pp. 1581-1586.

[30] T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss:, "Self organizing robots based on cell structures – CEBOT," in *IEEE International Workshop on Intelligent Robots*,

Piscataway, 1988, pp. 145-150.

[31] K. A. Hawick, H. A. James, J. E. Story, and R. G. Shepherd, "An Architecture for Swarm Robots," School of Informatics University of Wales, North Wales, UK, Technical Report DHPC-121., 2002.

[32] M. Gupta and K. Singh, "AutoBot: A Low Cost Platform for Swarm Research Applications," in *3rd International Conference on Emerging Trends in Engineering and Technology*, DC, USA, 2010, pp. 33-36.

[33] H. Wei, Y. Cai, H. Li, D. Li, and T. Wang, "Sam-bot: A Self-assembly Modular Robot for Swarm Robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, 2010, pp. 66-71.

[34] M. Dorigo, "SWARM-BOT: An experiment in swarm robotics," in *IEEE Swarm Intelligence Symposium*, Los Alamitos, CA, 2005, pp. 192-200.

[35] M. F., et al., "Swarmbot: From concept to implementation," in *International Conference on Intelligent Robot and Systems*, Piscataway, NJ, USA, 2003, pp. 1626-1631.

[36] J. McLurkin and D. Yamins, "Dynamic Task Assignment in Robot Swarms," in *Robotics: Science and Systems*, Cambridge, USA, 2005.

[37] G. Caprari and R. Siegwart, "Mobile micro-robots ready to use: Alice," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Alberta, Canada, 2005, p. 3295–3300.

[38] W. C. Evans, G. Mermoud, and A. Martinoli, "Comparing and modeling distributed

control strategies for miniature self-assembling robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, AK, USA, 2010, pp. 1438-1445.

[39] C. M. Cianci, X. Raemy, J. Pugh, and A. Mar, "Communication in a Swarm of Miniature Robots: The e-Puck as an Educational Tool for Swarm Robotics," in *Simulation of Adaptive Behavior, Swarm Robotics Workshop, Lecture Notes in Computer Science (LNCS)*, 2006.

[40] G. Lee and N. Y. Chong, "Low-Cost Dual Rotating Infrared Sensor for Mobile Robot Swarm Applications," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 277-286, Mar. 2011.

[41] H. Surmann, A. Nüchter, and J. Hertzberg, "An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments," *Robotics and Autonomous Systems*, vol. 45, no. 3-4, pp. 181-198, Dec. 2003.

[42] K. Lee and W. K. Chung, "Effective Maximum Likelihood Grid Map With Conflict Evaluation Filter Using Sonar Sensors," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 887-901, Jul. 2009.

[43] F. Rivard, J. Bisson, M. F., and D. Letourneau, "Ultrasonic relative positioning for multi-robot systems," in *IEEE International Conference on Robotics and Automation*, Pasadena, CA, 2008, pp. 323-328.

[44] F. Mondada, et al., "SWARM-BOT: From Concept to Implementation," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, USA, 2003, pp. 1626-1631.

[45] A. Takeshi, M. Yuki, and H. Shigeo, "Mechanical Design of Three-Wheeled Lunar Rover," in *IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[46] H.-C. Huang, "Intelligent motion controller design for four-wheeled omnidirectional mobile robots using hybrid GA-PSO algorithm," in *IEEE International Conference on Systems, Man, and Cybernetics*, Anchorage, AK, 2011, pp. 2267-2272.

[47] A. Seeni, B. Schäfer, and G. Hirzinger, "Robot Mobility Systems for Planetary Surface Exploration – State-of-the-Art and Future Outlook: A Literature Survey," in *Aerospace Technologies Advancements*. Vukovar, Croatia: Intech, 2010, ch. 10, p. 492.

[48] W. Zhidong and V. Kumar, "Object closure and manipulation by multiple cooperating mobile robots," in *Proceedings IEEE International Conference on Robotics and Automation (Volume:1)*, 2002, pp. 394-399.

[49] R. M. Voyles, "TerminatorBot: a robot with dual-use arms for manipulation and locomotion," in *Proceedings IEEE International Conference on Robotics and Automation*, vol. 1, San Francisco, CA, 2000, pp. 61-66.

[50] M. Wang, S. Ma, B. Li, H. Gong, and Y. Wang, "Reconfiguration optimization for a swarm of wheel-manipulator robots," in *IEEE International Conference on Mechatronics and Automation* , Takamatsu, 2008, pp. 1035-1040.

[51] L. Vega, et al., "MILyBots: Design and Development of Swarm-Robots," in *Florida Conference on Recent Advances in Robotics*, Melbourne, Florida, 2008, p. 119–122.

[52] M. B. Dias and A. (. Stentz, "A Market Approach to Multirobot Coordination," Robotics Institute, Carnegie Mellon University Technical CMU-RI -TR-01-26, 2001.

[53] M. Li, et al., "Robot swarm communication networks: Architectures, protocols, and applications," in *Third International Conference on Communications and Networking*, China, 2008, pp. 162-16.

[54] B. Samanta, C. Nataraj, S. Reddy, M. Woods, and C. Nataraj, "SWARM ROBOTICS: Research Experiance for High School Students," in *ASEE Midatlantic Conference*, 2009.

[55] F. Arvin, K. Samsudin, and A. R. Ramli, "Development of a Miniature Robot For Swarm Robotic Application," *International Journal of Computer and Electrical Engineering*, vol. 1, no. 4, pp. 436-442, 2009.

[56] C. Churavy, et al., "Effective implementation of a mapping swarm of robots," *Potentials, IEEE* , vol. 27, no. 4, pp. 28-33, Aug. 2008.

[57] A. E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 97-120, Dec. 2008.

[58] F. Mondada, et al., "The e-puck, a Robot Designed for Education in Engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, Castelo Branco, Portugal, 2009, pp. 59-65.

[59] M. Dorigo, et al., "Swarmanoid: a novel concept for the study of heterogeneous robotic swarms," Institut de Recherches Interdisciplinaires et de D´eveloppements en

Intelligence Artificielle Technical TR/IRIDIA/2011-014, 2011.

[60] M. Patil, T. Abukhalil, and T. Sobh, "Hardware Architecture Review of Swarm Robotics System: Self-Reconfigurability, Self-Reassembly, and Self-Replication," *ISRN Robotics*, vol. 2013, p. 11, Apr. 2013.

[61] E. M. H. Zahugi, A. M. Shabani, and T. V. Prasad, "Libot: Design of a low cost mobile robot for outdoor swarm robotics," in *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, Bangkok, 2012, pp. 342-347.

[62] P. Moubarak and P. Ben-Tzvi, "Modular and reconfigurable mobile robotics," *Journal of Robotics and Autonomous Systems*, vol. 60, no. 12, p. 1648–1663, Sep. 2012.

[63] C. W. Zecha, J. Link, and W. Claupein, "Mobile sensor platforms: categorisation and research applications in precision farming," *Journal of Sensors and Sensor Systems*, vol. 2, no. 1, pp. 51-72, Jan. 2012.

[64] C. Churavy, et al., "Effective Implementation of a Mapping Swarm of Robots," *IEEE Potentials*, vol. 27, no. 4, pp. 28-33, Jul. 2008.

[65] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile Robot Positioning & Sensors and Techniques," *Journal of Robotic Systems (Special Issue on Mobile Robots)*, vol. 14, no. 4, pp. 231-249, 1997.

[66] J. Borenstein and F. L., "UMBmark: A Benchmark Test for Measuring Odometry Errors in Mobile Robots," in *Proceedings of the SPIE Conference on Mobile Robots*,

Philadelphia, 1995, pp. 22-26.

[67] F. Ducatelle, G. A. D. Caro, C. Pinciroli, F. Mondada, and L. M. Gambardella, "Communication assisted navigation in robotic swarms: self-organization and cooperation," Dalle Molle Institute for Artificial Intelligence Technical IDSIA-01-11, 2011.

[68] M. S. Couceiro, C. M. Figueiredo, J. M. A. Luz, N. M. F. Ferreira, and R. P. Rocha, "A Low-Cost Educational Platform for Swarm Robotics," *International Journal of Robots, Education and Art*, vol. 2, no. 1, pp. 1-15, Feb. 2012.

[69] T. Y. Abukhalil, M. Patil, and T. Sobh, "Comprehensive Survey on Decentralized Modular Swarm Robotic systems and Deployment Environments," *International Journal of Engineering (IJE)*, vol. 7, no. 2, pp. 44-73, Jun. 2013.

[70] H.-G. Beyer and U.-M. O'Reilly, "Emergence of Communication in Competitive Multi-Agent Systems: A Pareto Multi-Objective Approach," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, Washington, DC, USA, 2005, pp. 51-58.