GUIDELINES FOR BUILDING EXPERIMENTAL MOBILE ROBOTS WITH OFF-THE-SHELF COMPONENTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALİ GÜRCAN ÖZKİL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

FEBRUARY 2008

Approval of the thesis:

GUIDELINES FOR BUILDING EXPERIMENTAL MOBILE ROBOTS WITH OFF-THE-SHELF COMPONENTS

Submitted by **ALİ GÜRCAN ÖZKİL** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan ÖZGEN ____________________
Dean, **Graduate School of Natural and Applied Sciences**

Prof. Dr. Kemal İDER ____________________
Head of Department, **Mechanical Engineering**

Assoc. Prof. Dr. Buğra KOKU ____________________
Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Reşit SOYLU ____________________
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Buğra KOKU ____________________
Mechanical Engineering Dept., METU

Asst. Prof. Dr. İlhan KONUKSEVEN ____________________
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Yiğit YAZICIOĞLU ____________________
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Erol ŞAHİN ____________________
Computer Engineering Dept., METU

Date: 8.2.2008

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ALİ GÜRCAN ÖZKİL

Signature:

ABSTRACT

GUIDELINES FOR BUILDING EXPERIMENTAL MOBILE ROBOTS WITH
OFF-THE-SHELF COMPONENTS

ÖZKİL, ALİ GÜRCAN

M.S., Department of Mechanical Engineering
Supervisor: Asst. Prof. Dr. Buğra KOKU

February 2008, 86 pages

Robotics is an emerging field, and it is also affecting several other fields. Design of robotic platforms gains more importance since the focus and aim of the robotics research broadens widely and the variety of the users is significant. This work aims to present the design of a modular mobile robotic platform, which should be simple, easy to build and easy to use. The concept of modularity, usage of off-the shelf components and utilizing a PC platform, are addressed in this work. As a result of this work, a conceptual design is presented, and a prototype is built to highlight some important aspects of the conceptual design.

Keywords: Modular Robots, Mobile Robots, Robotics Design, Mechatronics, Configurable Systems

ÖZ

HAZIR KOMPONENTLER İLE DENEYSEL HAREKETLİ ROBOT YAPIMI İÇİN YOL HARİTASI

ÖZKİL, ALİ GÜRCAN

Y.Lisans, Makina Mühendisliği Bölümü
Danışman: Yrd. Doç. Dr. Buğra KOKU

Şubat 2008, 86 sayfa

Robotik, pek çok diğer araştırma konusunu da etkileyerek hızla gelişen bir dal olmuştur. Robotik platformların tasarımlarını önemi, robotik araştırmalarının ve kullanıcı profillerinin çeşitliliklerinin artmasına paralel olarak artmaktadır. Bu tez çalışması, basit, kolayca oluşturulabilen ve kolayca kullanılabilinen bir robot platformunun tasarımını amaçlamaktadır. Modülerlik kavramı, kolayca bulunabilinen bileşenlerin kullanılması, ve kişisel bilgisayar tabanlı olması, belirtilen konuları ele alırken kullanılan yöntemlerdir. Bu tez çalışmasının sonucunda, kavramsal bir tasarıma ulaşılmış; ve bu tasarımın önemli noktalarını göstermek amacı ile bir prototip üretilmiştir.

Anahtar kelimeler : Modüler Robotlar, Mobil Robotlar, Robot Tasarımı, Mekatronik Tasarımı, Ayarlanabilen Sistemler

To My Family and Friends

ACKNOWLEDGMENTS

The roots of this thesis work lie back to our discussions throughout the Mechatronics Design course in 2004. Together with Dr. Bugra Koku and Onur Yarkinoglu, we aimed to make an attempt towards modular mobile robot design-

I would like to express my deepest gratitude to Dr. Bugra Koku, for his direction, encouragement, patience and sincerity. He not only guided me about this master's thesis, but also shared his thoughts and advised me on my personal and academic decisions. This work could not be possible without him.

I am thankful to Dr. Zhun Fan and Dr. Torben Sørensen from Technical University of Denmark, whom believed in the potential of this master thesis, and provided some of the hardware used in the prototype phase. I am also grateful to Dr. Steen Dawids from Bispebjerg Hospital, Copenhagen, whom showed an appreciation to my work and envisioned the possible uses of modular mobile robot systems in hospital environments.

To my friends, Hakan, Ozgu, Ender, Baris, Mads, Josefin, Alvise, Jean, Lene, Anders, Thor and many others; I owe them a lot for their support, suggestions and friendship. They were always there when the nights were dark and long; helping with some paperwork, brewing an extra cup of coffee, or offering a piece of chocolate.

Finally, I would like to thank to my mother, my father and my sister, for being my family...

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The aim of this work is to propose a design of a modular mobile robot platform, with an intention for academic use.

The objectives can be listed as follows:

- To design a robotic platform which is inexpensive, easy to use and easy to maintain
- To propose a modular product architecture as a tool to provide flexibility, easiness and maintainability as claimed
- To adopt a PC based platform and investigate the advantages of using this approach
- To use off-the-shelf components whenever possible and reveal the benefits of this approach
- To show the development potential of such a system in terms of available software libraries and drivers, and hardware peripherals and resources using standardized hardware interfaces.

1.1 Definitions

1.1.1 Modularity

Modularity is the functional decomposition of a system, in smaller functional units. Modularity is an issue in both hardware and software in robotics. Software modularity has already been achieved through object oriented programming languages which brings componentized approach to software development.

Hardware modularization, on the other hand, is more about grouping of components that creates specific functions and creating forms that permits these functionalities. Therefore, hardware modularity is bound with physical constraints that come from components and physical interactions with users and environments.

1.1.2 Interface

A module is a singular functional unit of a system, and interface is the glue layer between modules, which provides connectivity and abstraction. Mainly, it permits the flow of energy and information through modules.

1.1.3 Modular Design

There are two different product architecture types; integrated architectures and modular architectures while designing a product. In the context of this thesis, modular design is referred to some of the key features of the proposed robotic platform. Flexibility and maintainability are the most important of these features.

As the designed platform is intended to be used in a wide range of educational activities, it should be easily reconfigured to different tasks. Modular design will provide different modules to accomplish the flexibility needed. In addition, if certain module of the robot malfunctions, it should be simple to replace, or at least disable that module such that the robot and the remaining modules will be functional.

1.1.4 Ease of Use

Ease of use is a rather general and relative concept. Therefore, it is needed to be explained in the context of this thesis. Firstly, it is related to the resources available on the platform. In order to be easy to use, the platform should provide certain means of development environments. As a very important portion of robotics work is about software development, it should provide sharing and reusability of code. The availability of software libraries to support development in various levels is closely related to the ease of use. This condition implies the importance of standardization, and using off-the-shelf components in the design is partially for this reason. For instance, using commercially available webcams for vision brings up the possibilities of the use of highly available drivers and image processing libraries, as opposed to specialized and robotics vision hardware with limited resources.

1.2 Introduction

Even though it is difficult to give a precise definition, a machine which can sense its environment, manipulate things, show some degree of intelligence and appears to have an intent or agency can be referred to as a robot [1]. The typical use of a robot is substituting human in dull, dirty or dangerous tasks.

From a technical perspective, International Federation of Robotics (IFR) classifies robots as: Industrial robots, professional service robots and personal service robots [2]. Being used in all modern manufacturing settings, industrial robots achieved a significant degree of success in the last couple of decades and came to a maturity. On the contrary, compared to industrial robots, service robots are still far away from having a significant place and usage in everyday life. Yet, a United Nations survey claims that the situation will change in the near future, and there will be a rapid growth in service robotics market, due to the demand from the various levels of society (market pull) [2]. It is therefore foreseeable that mobile robotics will be more and more commonly used and take a central part in service robotics market.

Research in mobile robotics is very broad and ever-evolving. As the field is emerging, robotics education is subsequently affected by the breadth of the topics covered and the depth of studies in these topics. It is essential for robotics education to be supported by suitable experimental exercises, in order to facilitate learning by supporting theoretical knowledge with practice. The importance of availability of robotic systems, and related hardware and software resources to support education, and therefore, the need for a simple, cheap and easy to use mobile robotic system is obvious.

As a result of these considerations, this thesis work aims to outline a design for such a system for academic and educational use. An important characteristic of this work is the design methodology; using a product design perspective rather than a solely technical approach.

Regarding this plan, it is necessary to have a user-centered design and reflect the needs of typical users in the system of such an educational mobile robot. The ease of use is the vital consideration, since the users of the system would have different backgrounds and experience. As robotics cover a large spectrum of subjects, this system should provide diverse resources in terms of hardware and software, yet it should be easy to learn and simple to operate.

From conceptual perspective, the aspects of the design can be characterized as modularity, flexibility and “easiness”. Modularity is the key aspect that enables the requirements of the design. By analyzing and abstracting different needs and tasks of a mobile robot, the complex design problem can be divided into a set of simpler problems and dealt with individually. This approach induces the design work to module level, and provides the opportunity to evolve or vary these modules independently, since they provide the same general functionality. It is then truly possible to imagine the system in an abstract level.

Flexibility is also closely related to the modularity of the system. The overall system performance depends on how individual modules create a whole. In a flexible system, different sets of modules might come together and create a specific system depending on different needs. It is therefore important to carefully define the interfaces between the modules of the system.

The final characteristic of the design is easiness. Being rather general, this requirement should be reflected at various levels of the design. The system should be consisted of simple, easily accessible and affordable elements. These elements should be easy to obtain, operate, install and maintain. It is therefore beneficial to use off-the-shelf components as much as possible. Consequently, the mechanical structure of the robots should be 'easy'. Accessing and maintaining the modules, components inside the modules and the whole system depends on the design of mechanical structure. And finally, the control system of the robot system should be 'easy'. It should provide simple and convenient interfaces to its physical resources and provide a simple platform for software implementation and development.

In short it is believed that the proposed design approach to the field of mobile robotics will result in a simple, affordable and easy to use system which can enhance learning progress in robotics education.

CHAPTER 2

STATE OF THE ART

Robotics is a popular research topic especially for last two decades. Due to the multidisciplinary nature of the field, it is possible to see robotics projects in different departments of universities, such as electrical, mechanical, mechatronics, aeronautical engineering or computer science. Also, due to fiction, literature and the practical nature of the field, robotics is also becoming a popular hobby. There are a number of amateur clubs, where people build robots, share information and arrange events for promoting robotics. And finally, there are a growing number of commercial applications of mobile robotics, where solutions not only focus on technical problems, but also user related issues. In short, there are a number of people and organizations that need, use and/or design robots for different reasons and before proceeding with the design phase, it is necessary to review some of these works with similar purposes. The following sections give the details of these reviews, and a summary is given in Table 1 and Table 2.

2.1 Modular Robots

One of the features of the design proposed in this work is modularity. A modular design is generally characterized by the functional partitioning into discrete and reusable modules consisting of self contained functional elements [3]. Naturally, there have been several attempts to a modular design of a robot. Robo-cup, designed in Graz University, Austria, is one of the noteworthy robots for its modularity [4]. It was used for robotic soccer events, and it was designed for reuse of the components, exchangeability of modules and flexibility of the overall system.

Using a layered modular approach, it had an omni-directional driving layer, a range finder layer consisting of a laser range finder and 24 sonars, a controller layer consisting of a single board PC with wireless LAN and CAN bus, and a vision layer consisting of an omni-directional camera. It is a very good example for its modularity, but its design is more focused on a single task, and it has relatively large physical dimensions. (Figure 1)

Figure 1: Robocup robot, University of Graz [4]

Another modular design presented in the RABBIT platform, which in general deals with mechatronic systems. Its aim is to develop a modular hardware and software platform for distributed real-time applications. Sharing the same basic idea, decomposition of a mechatronics system into subsystems, RABBIT platform aims to achieve high flexibility and extensibility for both hardware and software components [5]. It focuses on the real-time performance of the system, and features field programmable gate-logic arrays (FPGA) as a central component in each module. As a case study of a mechatronic system, it has been prototyped on a mobile robot called X-mobile (Figure 2).

Next robot reviewed is from University of Tennessee, which is designed for under-vehicle inspection purposes. Named as "Brick architecture", this robot consists of five logically and physically independent modules; data acquisition, data processing, data transmission, power and self-management modules [6]. Although it utilizes a modular structure, the aim is to only make it easier to design the software, and manufacture and maintain the hardware, rather than having a re-configurable platform.

As mentioned, amateurs and hobbyists are also contributing to mobile robotics. BotStack is a good example for that, utilizing several layers with different uses for robot competitions [7] (Figure 3). It is an attempt to achieve a simple distributed computing system as well, by utilizing microcontrollers in each layered module. A similar feature of the system is that the modules are also designed for physical stackability, using pin headers. It is more of modularity in electrical hardware rather than complete robot, but it clearly exemplifies the advantage of using such an approach even in amateur robotics.

Figure 2: X-mobile robot [5]

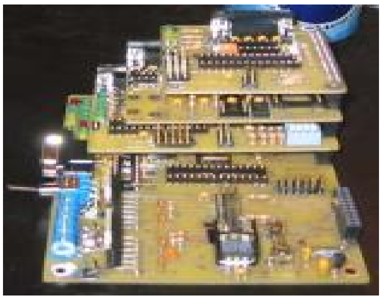


Figure 3: BotStack architecture [7]

2.2 PC Based Robots

In addition to modularity, another feature of the design is the use of a PC mainboard as the control system. Because it provides a lot of processing power on a familiar developing environment inexpensively, standard computer systems are becoming more and more popular every day. One of the first examples is the Open Automation Project [8] (Figure 5). Its main goal is to achieve an intelligent PC-based mobile robot for office and home use, through utilizing software and electronic components. It is a community supported open-source project, therefore the design blueprints and the software is freely distributed. One of its goals is very related to design proposed in this work; using consumer level components instead of professional components and minimize the cost of the system for around a price of a PC (~$1500).

Centibot's project [9] also uses PC mainboards as a controller unit. Based on commercially available Amigobot [10], it uses a mini-itx mainboard, wireless LAN and webcam as the hardware platform. The aim of the project is to demonstrate coordinated intelligence of 100 robots in an urban environment. The project involves mapping and tracking objects, therefore require extensive amount of processing and data sharing, which is very relevant to the choice of a PC based structure.

Another interesting use of a small form factor PC in robotics is demonstrated in University of Oklahoma [11]. In this case, a Mac computer was used as the controller unit on a commercially available Pioneer-AT robot [12] (Figure 4). Being a very small form factor, fully functional computer, Mac Mini was used for processing laser scanner data and firewire camera feed. The project exhibits the simplicity of using a computer on a robot, as the computer provides processing, data storage, interfaces and communication functionalities as a whole in a small package.

Figure 5: OAP robot [8]

Figure 4: Mac-Mini robot [11]

2.3 Commercially Available Educational Robots

2.3.1 Kpehera

Khepera is a research and education robot which is available from K-team Company [13] (Figure 6). It is principally designed for swarm robotics experiments, thus it is optimized for multi-robot applications. Latest version of Khepera robot base brings very high resolution encoders, ultrasonic and infrared range finders, and swappable batteries for continuous operation. Additionally, Khepera architecture provides extensibility using a modular approach. Currently following modules are available for the Khepera robot base:

Korebot: Korebot is a miniature single board computer platform, which utilizes an embedded CPU similar to the ones on hand-held devices. It provides various standard interfaces including USB, RS232 and Compact Flash connectors. It is possible to run an embedded Linux distribution on this board, thus it is offers a number of connectivity and development solutions. It is therefore possible to use USB cameras, mass storage, or WiFi/bluetooth adapters with Compact Flash interfaces; if these devices are supported by the operating system.

In order to extend the usage of Korebot from simple applications, it is needed to use some accessory boards such as motor driver, connectors, audio input/output, and analog/digital input/output boards.

Vision modules: For the previous version of Khepera (II), four different vision modules were developed.

K2D video turret: It provides a high resolution camera and it requires an off-board frame grabber with a cable connection to the robot. There is also a version with a spherical mirror. Major disadvantage is the tethered operation and the need for an external processor.

Wireless video turret: It is basically an adaptation of a wireless security camera to the Khepera II base. It outputs analog video; therefore an external processor and an image capturing card are needed.

Linear Vision Turret: It brings up a 64 pixel linear vision sensor which can be used for light intensity sensing. It is developed for soccer playing Khepera robots, which makes it possible to detect landmarks and identify light sources.

Matrix vision module: This module is an adaptation of CMU Cam module, which utilizes an 176x255 pixel sensor and embedded processor that can handle simple tasks like blob tracking.

Radio module: It brings up RF serial connection in 2.4 GHz band, with up to speeds of 115200 bps. It makes it possible to connect to and control seven different robots at the same time using this module.

Gripper Module: It basically utilizes two DC motors with absolute position encoders; for object manipulation and transportation. It also detects the size and electrical resistivity of gripped objects.

In conclusion, Khepera is a miniature modular robotic platform, which is mainly designed for swarm robotics research. It has some disadvantages compared to the design proposed in this thesis. First of all, Khepera was designed for a rather specific field of research in robotics. The miniature size of the robot is a great advantage in swarm robotics, but it is not very critical for most of the robotics education activities. Miniaturization affects the possible extensibility of the hardware and limits the development and it very much affects the cost of the robot.

The modular concept in the Khepera platform is also different from the modularity in this thesis. Khepera utilizes rather a hybrid approach, as the robot base is an integral design which features actuators, sensors and power on the same package. The other modules are provided as accessory products, which enhances the capabilities. In the proposed design, it is aimed to decompose sensors, actuators and controllers completely and achieve singular functionality in each module.

With the latest version of the platform, it is now supported with an embedded PC which widens the capabilities of the platform greatly. Still, due to the size limitations, the computer used is not very capable, and due to the specific operating system to be used, there are many limitations on the external devices that can be used. Additionally, the software support is also not as extensive as it is on a regular personal computer system. It is also needed to use accessory boards in order to provide even some standard interfaces. Finally, due to this specific hardware, it is very expensive compared to the mini-itx platform that is used in the proposed design.

Figure 6: Khepera III Miniature Robot Base [13]

2.3.2 Lego Mindstorms

Lego offers the Mindstorms NXT toy robotics system [14] (Figure 7). In addition to the traditional structural components Lego manufactures, this system offers a central controller brick, a number of sensors and actuators. The central brick houses a reprogrammable microcontroller and embedded Bluetooth connectivity, and it provides connectors for four sensors and three actuators. Sensors and actuators can be monitored, and the controller brick can be remotely programmed using a graphical programming language.

In terms of actuators, only component available is the servo motor unit with built-in encoders.

In terms of sensors, standard components are a bumper sensor, an ultrasonic sensor, a microphone and a light intensity sensor. In addition to these components, there are also some accessory sensors such as an accelerometer, a gyroscope, a temperature sensor and a color sensor. Lego provides the interface standards for the controller brick, so new sensors and accessories are being designed independently by various companies and organizations.

Figure 7: Lego Mindstorms NXT Robotic System [14]

Lego Mindstorms system adopts a componentized approach rather than a modular approach. In this approach, functional decomposition is induced to a more specific level. Basically, it is related to the degree of decomposition of functions. To exemplify, a motor can be defined as a component; and several motors, together with other components, might constitute modules, which provides singular functions; such as locomotion. Therefore, It is possible to achieve an endless number of different configurations using the same set of components, but the achieved robots actually feature an integrated architecture, as the components of the system are structurally bundled together in a specific manner.

To compare the Lego Mindstorms with the design proposed in here, even though both systems offer reconfigurability, the level of modularity are different. As mentioned, Lego system is component based whereas this thesis uses a module based approach. To exemplify; Lego offers motors as components, whereas the proposed design offers I.e. Locomotion modules, which can be composed of several motors and other components. Therefore, in order to provide movement to a Lego robot, it is needed to program the actuation of these motors individually for different tasks, whereas a locomotion module might only need an abstract movement command.

The availability of resources is also a significant limitation in the Lego system. There only are four ports available to connect sensors. Sensors are also singular; if it is needed to detect ranges of objects, up to four ultrasonic sensors can be connected and they cannot cover the whole circumference.

Capabilities of the brick controller are also limited since it is a simple micro-controller based system. As it is designed for a very wide range of users; it is very simple to physically build the systems and program. It provides a similar software interface to some of the components as the proposed design, as serial communications are used to get or set properties of these components. Sensors such as the bumper and the microphone do not provide this functionality, which compromises the functional decomposition of tasks. This also affects maintainability, since if the digital or analog input chips of the brick gets damaged; it is needed to replace the whole controller brick, which is very costly.

2.3.3 VEX Robotic System

Vex robotics system is very similar to the LEGO system in principle [15] (Figure 8). It mainly focuses on mechanical hardware and it provides various structural components for building different robots. It provides a centralized microcontroller which can be programmed offline, and it offers a limited number of sensors such as ultrasonic range sensors, shaft encoders and bumpers. Compared to the Lego system, it is sturdier as it uses metallic parts, but its' software resources are limited.

As it is in the case of Lego Mindstorms, it is a reconfigurable system rather than a modular system, as the robots built has integrated product architectures. It is very incapable in terms of connectivity, maintainability and software support compared to the design proposed.

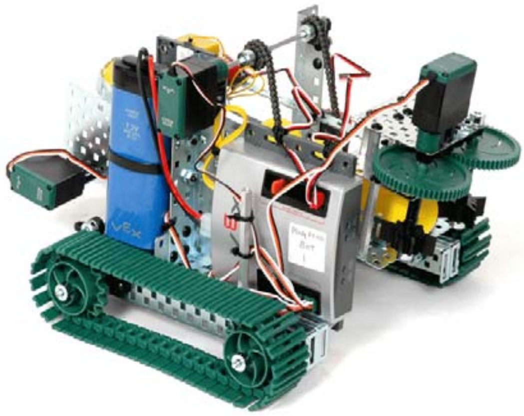


Figure 8: VEX Robotic System [15]

2.3.4 Parallax Kit

Parallax offers robotic and microcontroller kits mainly for educational use [16] (Figure 9). The robot kits are designed around Basic Stamp controllers, which provide interfacing to standard electronic components. These robots can be programmed using a Basic programming language. The robotic kit offers a very simple chassis, which houses the sensors, modified hobby servo motors and the controller board. The controller board has a small prototyping board, which is used to transfer pin connectivity to the Basic Stamp controller. Parallax offers a number of different accessories that can be connected to the controller board, but these accessories are mostly standard components with break-out boards that can be used on a prototyping board.

Parallax system is very much focused on educational use, but it is more focused on microcontrollers rather than robotics. As the company developed its products around their microcontroller design, the robot kits are extensible instead of modular. Although there are a number of different components provided by Parallax such as temperature sensors, accelerometers and range finders, these are singular components that needs to be interfaced to the Basic Stamp board.

As Parallax targets pre-undergraduate educational institutes, and the solutions provided are simpler and different than the modular design proposed in here. It is possible to experiment some very simple and introductory concepts using this kit.

Figure 9: Parallax BoeBot Kit [16]

Table 1: Features comparison of reviewed robots

ROBOT	**Modularity**	**Control Platform**	**Size**	**Power needs**
Robo-cup [4]	✓	Consumer PC	Large	High
RABBIT [5]	✓	FPGA	Large	High
Brick [6]	✓	Industrial PC	Large	High
BotStack [7]	✓	Microcontroller	Small	Low
OAP [8]	×	Consumer PC	Large	High
Centibots [9]	×	Consumer PC	Large	High
Mac –bot [11]	×	Consumer PC	Large	High
Khepera [13]	×	Industrial PC or Microcontroller	Miniature	Low
Lego [14]	✓ (componentized)	Microcontroller	Variable	Low
VEX [15]	✓ (componentized)	Microcontroller	Variable	Low
Parallax [16]	✓ (componentized)	Microcontroller	Variable	High

Table 2: Features comparison of reviewed robots (cont'd)

ROBOT	Flexibility	Expansion Potential	Availability (of buying or replicating)
Robo-cup [4]	Limited due to the physical shape	High due to the PC platform	Design outline is given, but no details are available
RABBIT [5]	High	Capable of expansion, but requires somewhat special hardware	No details are available on the design
Brick [6]	Low due to the single purpose application	Limited, as the modules are not identical physically	Design outline is given, but no details are available
BotStack [7]	High due to the possibility of multi-microcontroller architecture	Limited, due to the need of custom design of each layer	Design outline is given, but no details are available
OAP [8]	High, due to the usage of off-the-shelf components	High, due to the usage of off-the-shelf components	Open source hardware and software
Centibots [9]	Low, fixed application	High due to the PC platform, but available space for expansion is limited	Construction details are given with the parts needed
Mac –bot [11]	Low, fixed application	High due to the PC platform	Construction details are given with the parts needed
Khepera [13]	High, with the accessory turrets	Limited number of expansion options, but possible to custom design new layers	Commercially available. SDK and HDK are provided
Lego [14]	Very high	High, due to highly componentized approach. Several companies started to develop proprietary sensors for the kit	Commercially available. SDK and HDK are provided
VEX [15]	High	Limited, hardware expansibility is possible with the extra parts from manufacturer	Commercially available.
Parallax [16]	High	Limited, due to micro controller architecture, yet manufacturer provides several components	Commercially available.

CHAPTER 3

CONCEPTUAL DESIGN

To begin with the conceptual design, it is better to present the design criteria before proceeding with the details of the design procedure. These criteria are given as follows:

i. Usage of modular product architecture: As defined and addressed previously, this work aims to achieve modularity since it is believed to be beneficial.
ii. Usage of a PC-based platform: It is decided to use a mini-itx based platform as a controller for the robot. This requirement brings up several design constraints which are given in the following points.
iii. Usage of off-the-shelf components: It is claimed that usage of off-the-shelf components is advantageous in terms of availability, maintainability and inexpensiveness. Therefore, it should be kept in mind that commercially available, consumer level components and products should be modified and used in the robotic platform.
iv. Hardware modularity: It is required to achieve hardware modularity in terms of interfacing standards and stackability. Therefore, modules should be linearly stackable and provide identical hardware interfaces.
v. Software modularity: It is needed to provide simple and abstract software interfaces to available modules. The modules should simply send or receive meaningful information, so that low-level processing is dealt within the module itself. The functionality of the modules must be supported with a software library on the main controller module, in operating system level.
vi. Physical constraints: It is obviously needed to physically contain at least the biggest sized component in module level. In that case, as a mini-itx board is used, it is

needed to have an area of at least 17x17 cm. It is aimed to use the same physical structure in all modules to provide the standardization and physical stackability. Therefore, the footprint of the robot must be at least 17x17 cm, since the mini-itx board will be placed horizontal to the ground. In terms of the height of the modules, it is preferred to have the same height in each module, as it will be easier to manufacture.

In terms of power, it is needed to meet the different power requirements of the system. As the system is battery operated, it is needed to provide maximal operability with minimal added weight.

The weight of the system should be optimized, as it should be both balanced and easy to carry around.

It is needed to have a durable system since it is intended to be used in educational institutes. Material selection should be done considering both this fact and the weight considerations.

vii. Economical constraints: It is needed to minimize the cost of both development and the resulting system.

3.1 Exploration of Design Space

Before proceeding with the system design, it is needed to explore the commonly used components and solutions in robotics. It is required to review these components and solutions in a classified manner in order to correspond to the modular approach.

A mobile robot consists of three main components; sensors, actuators and controllers. Successful implementation of the robot therefore depends on both the individual performances of these components and their combined success. These three components individually represent a main class of modules, and it is possible to design different sub-classes of modules as long as they generically represent a main class. As addressed before, the classified modularity approach is a key aspect of the design, asserting the possibility of creating variations. Therefore, it is necessary to study these main classes and reveal the possibilities to see the design potential.

In addition, a number of additional components are used in mobile robotics for needs such as power, interfacing and communication. Following sections briefly outlines the major alternatives in locomotion, sensors, power solutions, hardware interfaces, controllers and communications.

3.1.1 Locomotion

Locomotion can be defined as the physical interaction between a vehicle and its environment. Movement is only possible with an action-reaction sequence, which is caused by the actuators and interaction forces generated by these actuators.

Mobility is a common feature for most of the agents in the nature. The locomotion principles of animals have been evolved to adapt to the variations in the environment. Some natural motion concepts that roots form these evolutions and that have been investigated in robotics are; crawling (of a worm), sliding (of a snake), running (of a cat), jumping (of a gazelle) and walking (of a human) [16]. Although these types of locomotion can be very robust and efficient in their respective environments, they are very difficult to imitate, mainly due to technical difficulties in actuation technologies. Instead, technical systems usually use wheeled or tracked systems, since they are more suitable for conventional rotary actuation.

While dealing with robot locomotion, there are three main issues to be tackled with; stability, contact characteristics and the type of environment [16]. The choice of locomotion method mainly depends on these considerations, and affects several system variables. First, the choice basically depends on or determines the number of actuators to be used. Consequently, the number and the physical characteristics of actuators affect the structural complexity of the robot. The configuration also influences the control expense of the locomotion. Effectiveness and operability of the robot depends on the power consumption of the locomotion system. And of course, terrain characteristics have a major consequence on the design of locomotion.

There are two main classes of locomotion that are being followed in robotics; which are legged locomotion systems and wheeled locomotion systems. Following sections summarize the basics of wheeled and legged locomotion for mobile robots.

3.1.1.1 Legged Locomotion

Legged locomotion is better suited for rough terrains. It can be preferable over wheeled locomotion, where the movement is needed on grounds with irregular characteristics. These irregularities might both exist in outdoor environments with naturally formed grounds, or indoor environments with manmade obstacles like doorsteps, gaps or stairs.

Leg is obviously the unit part of legged locomotion systems. A leg must have at least two degrees of freedom in order to provide mobility; one for lifting and one for swinging. Degrees of freedom of the leg entail the number of actuators as well; there generally exist one actuator per degree-of-freedom in this type of locomotion [16]. Most of the leg designs actually utilize three degrees of freedom, which result in better maneuverability of the robot. But this advantage comes with the cost of increased complexity in the structure, increased number of actuators and weight, and even more complicated control strategy.

Stability is an essential issue in legged locomotion. A mobile robot can be either statically stable or dynamically stable.

In general, static stability occurs, when the center of mass is completely within the support polygon, and the polygon’s area is greater than zero (The support polygon is drawn by connecting the ground contact points of the object). An explicit implication of this theorem is that a statically stable structure must have at least three ground contacts in order to have a non-zero area for the support polygon. Moreover, a mobile legged robot must have at least four legs to be able to feature static stability, since at least one of the legs will lose its contact with the ground during the motion. Most of the statically stable legged robots actually have six legs, since they can move faster by moving up to three of their legs at a time.

Other type of stability is mainly promoted by human-like robotics research, as human body is dynamically stable during locomotion. This type of robots usually has small footprints, which are even reduced to a single point or line of contact during walking. Active balancing of the body weight is therefore essential to prevent falling. Coordinating several degrees of freedom legs simultaneously, predicting the center of mass during accurately, and shifting the body weight to compensate inertial forces, is obviously quite a difficult and demanding task.

3.1.1.2 Wheeled Locomotion

Wheels are the most popular choices for locomotion in not only robotics but all kinds of ground vehicles. Ease of use and relatively high efficiency are the main factors for their extensive usage. Balancing is not as important as it is in legged locomotion, since the wheels already provide ground contact almost continuously.

Main considerations in this type of locomotion are traction, maneuverability and stability [17] . These issues are closely related to the characteristics of wheels.

A wheel is a circular device capable of rotating on its axis, facilitating movement or transportation or performing labor in machines [18]. An ideal wheel does not slip in the orthogonal direction of rolling, and no translation slip occurs between the wheel and the floor.

There are four main types of wheels that are being used in robotics. The first one is the standard wheel, which is described above. The standard wheel has two degrees of freedom, which will enable rotation around the axle, and the contact point. Motorizing the wheel around the wheel axle will yield one directional motion. The second type of wheel is the castor wheel, which also has two degrees of freedom. Main difference in a castor wheel is the off-set vertical axis of the wheel. This feature enables castor wheel to swing around its vertical axis. These two types of wheels are the most commonly used types in robotics; standard wheels are actuated and provide traction, and castor wheels are passively connected to the chassis and provide stability.

The other two types of wheels primarily differ in their degrees of freedom. A third degree of freedom in a wheel provides omni-directional motion, which is consequently related to maneuverability. Omni-wheel, or Swedish wheel, functions as a normal wheel if the motion is perpendicular to the actuation axis, but permits movement through the rollers attached around the wheel circumference if the motion is parallel to the actuation axis. The last type of wheels is the spherical wheel, which offers true omni-directional locomotion. Its configuration is very similar to ball-type computer mouse, where the friction between the ball and the rollers are converted to two dimensional coordinates. Although these types of wheels increases maneuverability, their realization is technically difficult compared to standard and castor wheels [16].

Wheel type and wheel arrangement must be considered concurrently while designing wheeled locomotion. Choice of the wheel type strongly relates to the possible arrangement of the wheels, and stability, controllability and maneuverability of the vehicles. Although static stability can be achieved with only two wheels, conventional systems utilize three or more wheels for practical reasons. Generally, stability can be improved by using more wheels, but it is then needed to utilize suspension to keep the ground contact in uneven terrains.

Maneuverability is also affected by both wheel type and configuration. Omni-directional movement provides the robots to move in a way, regardless of the initial orientation. Omni-directional locomotion is usually achieved by using Swedish or spherical wheels. In addition, castor wheels actuated in both vertical axis and rotating axle can be used to achieve omni-directional motion.

Differential drive configuration, which is the most popular choice in robotics, provides high maneuverability with a simple configuration. Using two driving wheels, the robot can rotate around its axle center, possibly without changing the footprint. In this configuration, the wheels are usually positioned close to the center of gravity, and stability is provided by one or more caster wheels.

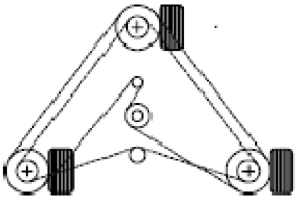
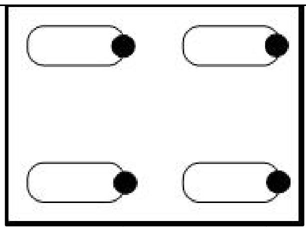
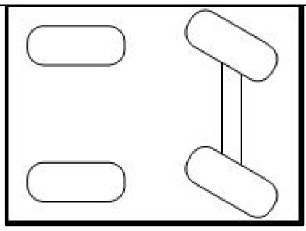
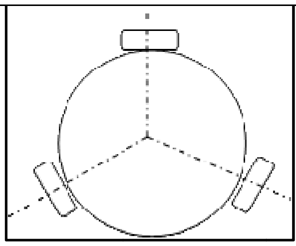
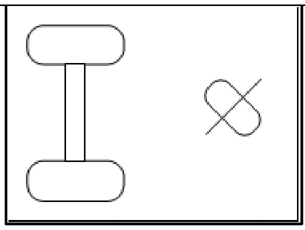
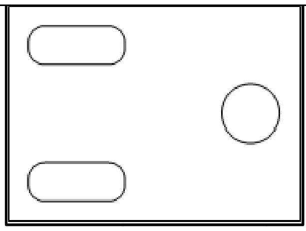
Ackerman steering, which is mainly used in automobiles is another configuration used in mobile robotics. It also needs two motors, but one is used for driving and one is used for steering. Maneuverability of the system is lower compared to differential drive, since the vehicle will need a turning diameter usually greater than its length. The advantage of the Ackerman steering is the increased stability in high speeds, especially while making turns.

Controllability is the next issue that should be considered. In an omni-directional configuration for instance, position and orientation of the robot will be determined by individual positional control of omni-wheels. Increased degrees of freedom at the wheel also affect the contact characteristics such as slippage, and result in less accurate positioning. In differential drive configuration, however, the position of a robot will simply depend on the positions of driven wheels. Ackerman steering is also easy to control, and is more accurate in certain cases such as moving along a straight line.

3.1.1.3 Synopsis of Locomotion

Stability, contact characteristics and the type of the environment are the main factors while deciding about locomotion. While legged locomotion provides better stability in rough terrains; it increases complexity, power consumption and the weight of the system. Since it is aimed to use the designed robot in indoor environments, wheeled locomotion seems to be more suitable for this case. Considering different types of wheels, and their arrangements; it is possible to say that the easiest way to implement locomotion in a mobile robot is by simple two-wheeled differential drive configuration. Consequently, prototype is going to be based on this type of locomotion.

Table 3: Wheeled Locomotion Configurations [16]

	*Synchro Drive. One motor controls the orientation of the wheels; another one is used for rotation.*	*Technically complex, robot orientation is not controllable*
	*Omni-drive with castor wheels. Rotation and orientation of each wheel is controlled.*	*True omni directional locomotion with improved stability. Requires two actuators per wheel*
	*Ackerman steering. Front wheels are steered together for turning. Front or rear wheels can be used*	*Increased stability, decreased maneuverability. Need of extra mechanism for steering.*
	*Omni-drive with Swedish wheels. Each wheel is driven.*	*Simplest omni-directional configuration. Difficult to compensate slippage, especially in uneven surfaces.*
	*Tri-cycle. Front wheel is steered, and rear wheels are driven together*	*Easy to control. Yet, there is still a turning radius.*
	*Differential drive. Driving wheels are driven separately.*	*Simplest mobility configuration. Usually supported with a third contact point. Straight line movement can be difficult.*

3.1.2 Sensors

3.1.2.1 Contact Sensors

Contact sensors are one of the simplest types of sensors that are used in robotics. Principally, they enable the detection or closeness of a physical contact.

The most commonly used contact sensors are mechanical switches, which open or close an electrical circuit. Due to their simplicity, mechanical switches are being produced in different shapes and size. They are very commonly used as bumpers in mobile robots, for detecting obstacles, or as limit switches if the mobile robot has a manipulator. It should also be mentioned that the human-machine interfaces of robots also make use of mechanical switches in forms of buttons.

Optical switches, such as light barriers, also act as contact sensors. They measure certain characteristics of light, like existence, infrared temperature or reflectance, and detect the closeness of objects. Unlike mechanical switches, optical switches are generally active sensors.

3.1.2.2 Locomotion Sensors

Locomotion is the key aspect of mobility in robotics. It is obviously important to sense the states of the elements of the locomotion system.

3.1.2.2.1 Wheel/Motor Position Sensors

By far, the most commonly used actuators used in robotics are DC motors, which provide rotary motion. Therefore, sensing the amount of rotation is essential in order to estimate the position, speed and acceleration characteristics of a system. Several techniques have been developed to sense the rotational movement. Some commonly used ones are; potentiometers which vary the electrical resistance, brush encoders which sense the contact of the brushes of the motor, optical encoders which sense the reflectance/existence of the light and magnetic encoders which sense the Hall effect using magnets attached to the shaft.

Among these alternatives, optical encoders are the most popular choice for measuring rotational data of motors and wheels. Acting as an optical switch, an optical encoder produces a certain sequence of square waves per revolution. The number of the waves defines the resolution of an encoder which is typically between 2000 – 10000 counts per revolution.

3.1.2.2.2 Heading Sensors

Heading sensors gives information about the orientation of the robot. Two most common types are compasses and gyroscopes.

Compasses basically detect the magnetic field of the earth to estimate the orientation of robot. Comparing the detected magnetic field with an initially calibrated value, a compass can give relative directional information. A major problem with compasses is the disturbance. Especially in indoor applications, compasses tend to be very unreliable due to factors like the structural steel, power lines, electronic objects, etc.

Gyroscopes, on the other hand, can give absolute directional information, because they use a fixed reference. Measuring angular momentum characteristics of a spinning flywheel, which is pivoted in three principal directions, it is possible to detect the heading characteristics.

3.1.2.2.3 Beacon-Based Sensors

The localization of a mobile robot usually has an essential application value [19]. Using active or passive beacons, it is possible to determine the position of the robot relative to those beacons. This determination process principally utilizes triangulation, which implies that at least two different beacons are needed for the estimation.

The global positioning system (GPS) is a navigation technology which was initially developed for military purpose, uses time-stamped radio communications with satellites orbiting around the world. Receiving signals simultaneously from typically more than 10 satellites, a GPS unit calculates the time of flight of the signals, determines the distance of the satellites, and estimates its relative position on earth. It is commonly used in outdoor robotics, but because it generally requires a clean line of sight to the satellites, the use of GPS is rather limited in indoor robot applications. Instead, local radio-based beacon systems are used in indoors. Similar to GPS, a radio unit receives signals from fixed stations with known positions in this case. An interesting application of this type beacons makes use of wireless Ethernet nodes in an indoor environment, providing both connectivity and position information at the same time [20].

In addition to radio based beacons, optical beacons are also used in indoors. Active optical beacon systems emit laser or infrared light, and the robot utilizes light sensors to estimate their distances and triangulate its position. Passive beacons usually have high reflectance characteristics, so the robot can emit light to environment, detect the reflectance and estimate the distances to these reflecting beacons.

3.1.2.3 Ranging Sensors

Gathering geometrical environment characteristics is one of the most important requirements in mobile robotics. Range sensors provide vital information for obstacle detection and avoidance. In addition, range sensors also commonly used for localization purposes by statistical matching of measurement.

The key aspect of range sensing is the estimation of the time of flight. The time passed between transmission and reception of a certain type of wave (with a known speed) will yield to the distance of from the reflecting object. Some important features of range sensing are; calculation of the exact time, the dispersal cone of the transmitted beam, the surface characteristics of the target object, and the speed of the mobile robot and the object.

3.1.2.3.1 Ultrasonic Range Sensors

First of the three most commonly used range sensors in robotics are ultrasonic sensors, which emit sound waves. Their effective range is approximately between 5cm to 500cm, and their typical resolution is about 2-4cm. They had been primarily developed for Polaroid cameras to enable auto-focus functionality, but also attracted the attention of robotics community [21].

A major property of ultrasonic range finders is the cone-like shape of the measurement range. As sound propagates in wave form, it is reflected from multiple points at the distant object. This is an advantage if it is only intended to detect if there is an object within the range of the measurement cone. But often, it is needed to have more precise information about the position of objects; therefore ultrasonic range finders can be disadvantageous. Another drawback of these sensors is their reliability depends on the acoustic reflectivity of the objects. Soft materials like fur, foam or cloth reflect the sound waves very poorly, confining the operability of these sensors. And finally, the speed is another drawback. The speed of sound is approximately 340 m/s in air at atmospheric pressure. Therefore, the time of flight and processing time in takes around 20ms to estimate the distance of an object at 3 meters [22]. Considering the facts that the mobile robot might be moving and it probably contains several ultrasonic range finders, the update cycle time restricts the moving speed of the robot for a reliable operation.

3.1.2.3.2 Infrared Range Sensors

These sensors use infrared light and measure the angle of the reflected light to estimate the distance of the object. They are relatively cheaper due to their less complex structures. Having an effective range up to 300 cm, they have a resolution about 1-3cm. Compared to ultrasonic range finders, infrared sensors typically have a much narrower beam. Therefore, it is very difficult to detect chairs, tables or similar overhanging objects; since the sensor should be directly facing the obstacle. Also, material reflectivity is again a problem. Transparent or semi-transparent materials like glass or acrylic are not easily detected by infrared sensors. Another drawback is the disturbances related to external infrared light sources. Sunlight, heat sources or other equipment using infrared light might interfere with the sensors, giving erroneous results.

3.1.2.3.3 Laser Range Sensors

Laser range sensors act similar to ultrasonic range sensors as they are based on time of flight, using structured light (laser) instead of sound waves. Their performance is much better compared to ultrasonic sensors due to the properties of laser light. Also called as lidars (light-radar), these sensors utilize a rotating mirror to cover an area.

The most commonly used lidar in robotics is the SICK LMS 200 [23], providing an angular resolution of 0.5 degree over 180 degrees of span. They can detect objects in a range of 5cm to 2000cm or more, depending on the reflectivity of objects. Scanning at a rate of 0.4 seconds per cycle, it is fast enough for most of the robotics applications.

An alternative lidar is also being used in robotics is the HOKUYO URG-04 [24]. Compared to SICK LMS 200, it has quite a lower range of detection, 2 to 400 cm; but it has a finer resolution and span (0.36 / 240 degrees). In addition, it is much smaller in size and weight, can communicate over USB, and requires less power; making it a strong competitor against SICK LMS 200.

The major disadvantage of lidar sensors is their price [25]. Their price tags do not enable them to appear the lower end of mobile robotics yet. But the increasing demand from the robotics community and industry shows that the lidar technology will be more affordable in the near future, as consumer robotics gains the expected momentum.

3.1.2.4 Vision Sensors

Most of the agents in the nature use vision as their primary source for perception. Based on light and reflectance, vision sense provides dense information about the environment.

Two dominating technologies used in robotics vision are charge coupled device (CCD) and complementary metal oxide semiconductor (CMOS) technologies. CCD sensors are very commonly used in digital imaging applications, and due to that there have been significant improvements on this technology in recent years. Almost all of the consumer digital cameras use CCD sensors capable of capturing 2048x1536 pixels or more. CMOS technology is more efficient than CCD in terms of power consumption and much cheaper to produce; but they lack the noise compensation capabilities of CCDs. Yet, they find a significant application area in digital imaging industry, such as in mobile phones, security cameras and military applications that require fast speeds.

Vision sensors are being used in mobile robotics in a wide extent. Some applications uses them as an accessory or in passive ways (i.e. Remote surveillance), whereas some of them highly relies on these sensors and image processing for operation (i.e. Darpa challenge contestants). Therefore, the needs from a vision sensor can be quite different depending on the application. For most of the robotic applications, simple webcams would provide sufficient performance whereas speed-critical applications might need more professional equipment with dedicated high bandwidth capturing equipments. An average webcam today, houses a 1.3 megapixels sensor, providing video in 800x600 resolutions. They almost exclusively support USB2 connectivity, with a capture rate of 15-30 frames per second.

3.1.2.5 Synopsis of Sensors

Sensors basically enable perceiving the environment or the internal state of the robot itself. They mostly deal with the physical contact or proximity of the objects, their relative positions, and the robot's position relative to a frame. Among the reviewed sensors, mechanical contact sensors, infrared range finders and CMOS cameras come into prominence due to their functionality, commonality and inexpensiveness. These sensors might work relatively simple individually, but coupling them with a PC based controller makes it possible to accomplish complex tasks.

In the prototype it is planned to use mechanical switches as bumper to provide physical feedback to the robot. In addition, infrared range finders are going to be used to detect objects and obstacles, and their relative positions to the robot. Finally a webcam is going to be used to enable visual perception, such like color detection, shape detection and object tracking.

3.1.3 Power Requirements

Mobility of vehicles highly depends on their power resources and consumption. For a mobile robot, power structure is crucially important to accomplish the task. Therefore, power issues have an important impact on the overall system design.

In order to have a maximal duration of operation, power needs and efficiencies of system components should be kept in mind during the design phase. It is also important to select components with similar or compatible power requirements to minimize the complexity of the power systems.

Almost all of the mobile robots use electrical batteries as their main power source. Size, number and capacity of these batteries strongly influence the mechanical design. In general, batteries tend to be the heaviest components in a mobile robot. Therefore, while designing for their placement, it should be kept in mind that their weight will affect on the locomotion characteristics and system stability.

In a mobile robot the following components require power:

- sensors
- control computer
- communication system
- locomotion system/actuators

The power need is evident especially for active type sensors. In case of ranging sensors, for instance, the power consumption will be even more significant since they require constant operation. A SICK lms200 lidar consumes 20W, whereas Devantech sonar will require around 1.1 W during operation.

Power consumption of the control computer mainly depends on the capabilities of the computer. If it is a processor intensive application, control computer might require heavy loads of power. In case of the mini-itx platform, the power demand might go up to 70 W [26].

Communication might very much depend on power as well. Especially in wireless communications, power consumption increases with the increasing distances. The type of the technology also plays an important role. Bluetooth standard, for instance, is low power as it optimizes the on-time of the radio.

Locomotion is generally a high power demanding part of a mobile robot. Actuators in this layer require power depending on their efficiencies, internal and external frictions. Since actuator characteristics may vary greatly, it is difficult to estimate the power consumption for locomotion. But, to give an idea, a standard hobby servo motor consumes about 0.3W while delivering its maximum torque [27].

Power is still the most hindering point of mobile devices. Although there are some highly anticipated fuel cell technologies appearing lately, battery technology being used in mobile robotics is very mature and has not changed for years.

Lead -acid batteries are the most commonly used type of batteries in mobile robots. They have mainly been developed for electrical vehicles, and as a result, they can supply large amounts of currents. Lead-acid batteries are usually very heavy compared to their energy capacity. But because they are very cheap and able to provide big currents, most mobile robots use lead-acid batteries as their power source.

Nickel based batteries are also used in mobile robotics. They have a much higher energy density compared to lead-acid batteries, but they have a shorter life span and they are relatively more expensive. Nickel-cadmium batteries typically have better cycle life than nickel-metal hydride types, but they lack the capacities a mobile robot typically need [28].

The last types of the batteries that are being used in mobile robotics are lithium based batteries. They provide relatively the highest energy density, but they need extra circuitry to limit voltage and current for safety reasons. They are typically used in notebook computers and cell phones due to their low weight, but have not been very popular in mobile robotics yet due to their high price.

3.1.4 Interfacing Options

Interfaces between components are a significant issue in a modular design. Both physically and in software level, stacking and connection of individual modules to each other and the main controller module should be as easy as possible.

Most important aspect of the module interfacing is the inter-communication protocol. As each module accommodates a local controller unit, which is in the form of a microcontroller based circuit board, a commonly supported communication protocol by these units and the PC mainboard should be facilitated.

There are several communication protocols being commonly used in mobile robotics, which basically root from electronics applications. RS-232 standard is the simplest and one of the most commonly used protocols. It permits asynchronous data transfer between two end points with speeds up to 115200 bits per second or more. It features a 9 pin DB-9, or less commonly a DB-25 connector, but in most of the applications, only 3 pins are used (Ground, Receive, Transmit). Most of the microcontrollers support this type of communication but in ttl voltage levels, so a level converter is generally necessary (typically a Max232). The disadvantage of RS-232 is that every connection takes up one physical port. As this standard is also being slowly abandoned in consumer electronics, modern computer peripherals and main boards generally does not support RS-232 anymore, or in case of main boards; it is only possible to find at most two RS-232 ports. Speed of communication is also very limited compared to other commonly used protocols.

Universal Serial Bus or USB is a similar protocol to RS-232, with more advanced features. It was designed to replace RS-232, and it is now the most commonly used interface in consumer electronics. Featuring a much simpler physical interface, USB 2.0 standard supports speeds up to 480 megabits per second. It provides plug-and-play functionality, enabling auto-detection of hardware, and it is also capable of transmitting power at 5V with currents up to 500mA. It works in a master-slave configuration, meaning that each hardware connected take up one physical port. Compared to RS-232, it is relatively easy to increase the number of physical ports, by using inexpensive USB hubs. Most main boards also provide pin connectors for extra USB ports.

In contrast to point-to-point connection characteristics of RS-232 and USB, there are some other protocols which support point-to-multi point serial communications.

One commonly supported protocol is Inter Integrated Circuit (I^2C), which was developed by Phillips Company. It uses two lines, one for data transmission and one for clock synchronization. A similar protocol, developed by Motorola, is the Serial Peripheral Interface (SPI), which uses four lines, one for clock, one for slave device selection, and two for transceiving. It is faster than I^2C and support simultaneous two-way communication. The common feature of these protocols is they permit attaching peripheral hardware as a daisy chain. Therefore it is possible for individual components of a system to connect to each other communicate over a single line. These protocols are commonly used in embedded electronic applications, but they are rarely supported by consumer level computer main boards. Yet, it is possible to find USB/I^2C or USB/SPI converting adapters.

3.1.5 Control System

Third and the most important component of a robotic system is its control system. Acting as a "brain", it coordinates system components, evaluates sensory information and executes actuation commands. Intelligence of the overall system is tightly related to the capacity and performance of this layer.

Hardware wise, there are several alternatives being used in mobile robots, depending on their complexity. Some robots use very basic systems with simple microcontrollers whereas some more advanced robots use a number of specialized computers simultaneously. The complexity of the task, autonomy, connectivity and the cost are major concerns.

One of the principal aims of this thesis work is to demonstrate the applicability of consumer level computers to robotics. Due to a great demand in IT industry, there is a big variety of computer products that are available for reasonable prices. In addition, there is an established support and availability in terms of software, which is a big advantage in terms of development. Therefore, it is favorable to use off-the-shelf computer systems instead of specialized robotics hardware.

The main computer system selected is a mini-itx form factor computer. The reason for this choice is the small form factor of the system. Measuring 17x17x5 cm, this platform provides a decent processing power and extensibility options in a very compact footprint.

In order to have an operational computer, this motherboard needs a minimal set of additional hardware. The first need is a medium for the storage of data. The most conventional type of computer storage is hard disks. In mobile robotic applications, 2.5” hard disks have been occasionally used. They are being used in laptop computers without requiring an external power source, and they are more durable against minor shocks and vibrations compared to 3.5” desktop computer hard disks. Alternatively, smaller form factor hard disks can also be used (which are found in portable media players and video recorders) but they are more expensive and provide less storage. Using a hard disk as a storage medium is easy and economical, but depending on the type of application and operating environment, endurance might be an issue; since hard disks have moving parts.

Solid-state storage provides better durability for robotics applications. Having no-moving parts and demanding significantly lesser power, they perform in harsh environments. Their most significant disadvantages over hard disks are their price and limited storage capacities. They also have a limited write cycles; a typical consumer level solid-state drive wears out after 100000-300000 cycles whereas industrial types can reach up to 5 million write cycles [29]. The most commonly used form of solid state storage is SD cards, which has been used in digital photography for many years. They are common and affordable. Solid state drives are still very expensive (8$ per GB, where as 0.25$ per GB in hard disks [29]), but a typical 2-4 GB drive will be adequate for most of the robotics applications. They are pin compatible with hard disks, but in order to use them in a personal computer a special interface adapter is needed.

Another requirement for the mini-itx mainboard is power. A motherboard requires several different voltage levels to work, while mobile robots usually run on fixed voltage level batteries. Therefore it is needed to use a DC/DC voltage converter in order to run the computer with robot batteries. A similar case is automotive computer applications, where several converters are already being sold.

And finally, it is needed to have means connectivity to the computer in order to be able to remotely control or observe. Although there are several wired and wireless communication alternatives, wireless Ethernet connection is obviously the most convenient method, in terms of speed, range and power requirements. As the motherboard does not have wireless Ethernet as a built in functionality, it is needed to use an external accessory for this purpose. The easiest solution is using a USB wireless adapter. They are compact and they provide flexible placing options by using a USB extension cord. But most of them feature an embedded antenna, which provides a limited range. If the range becomes an issue, PCI type adapters can be used, which permit upgrading of the antenna.

3.1.6 System Configurability

As several possibilities have been reviewed for sensors, actuators, control computers, power and communication, it is now possible to proceed with the design of the robot.

In terms of modularity, it is necessary to make a functional division and classification of modules of the robot. Most general functional division of components in a mobile robot can be given as; sensors, actuators and control computers. These also constitute the main classes of modules, and each class might be consisted of several elements, or sub-classes (Figure 10).

Sensor class has the most variety of sub-elements. It is because reliable operation of a mobile robot heavily depends on environmental perception. This perception might be in terms of distance measurements (Ranging sensors: Infrared, sonar, laser), visual sensation (Vision sensors: standard cameras, omni directional cameras, infrared thermal imaging), object detection (Bumpers, pyroactive infrared sensors, light barriers), audio detection (microphones, array microphones), or user interface elements (Touch pads, buttons, joysticks, remote controls), which can be classified as sensors.

Actuator class also consists of a number of sub-classes. Locomotion, by definition, is an obligatory class that should be used in a mobile robot. Wheeled and legged locomotion sub-classes constitutes several different elements like tracked locomotion, omni-directional locomotion etc. Effectors (pan-tilt devices, grippers, and articulated arms), displays (lights, screens) and audio devices (speakers, piezos) can be listed as other sub-classes of actuators.

Controller class basically includes means of orchestration of a mobile robot. PC-based controllers, as proposed in this design, represent a main subclass of controllers. As the form factors are continuously shrinking in computer industry, the base for the PC might be a mini-itx mainboard, a laptop or an ultra-mobile computer.

Creating a hierarchical tree of modules therefore enable the exploration of the design space. Using this information, it is possible to see different assemblies of modules or which modules should be used mutually exclusive.

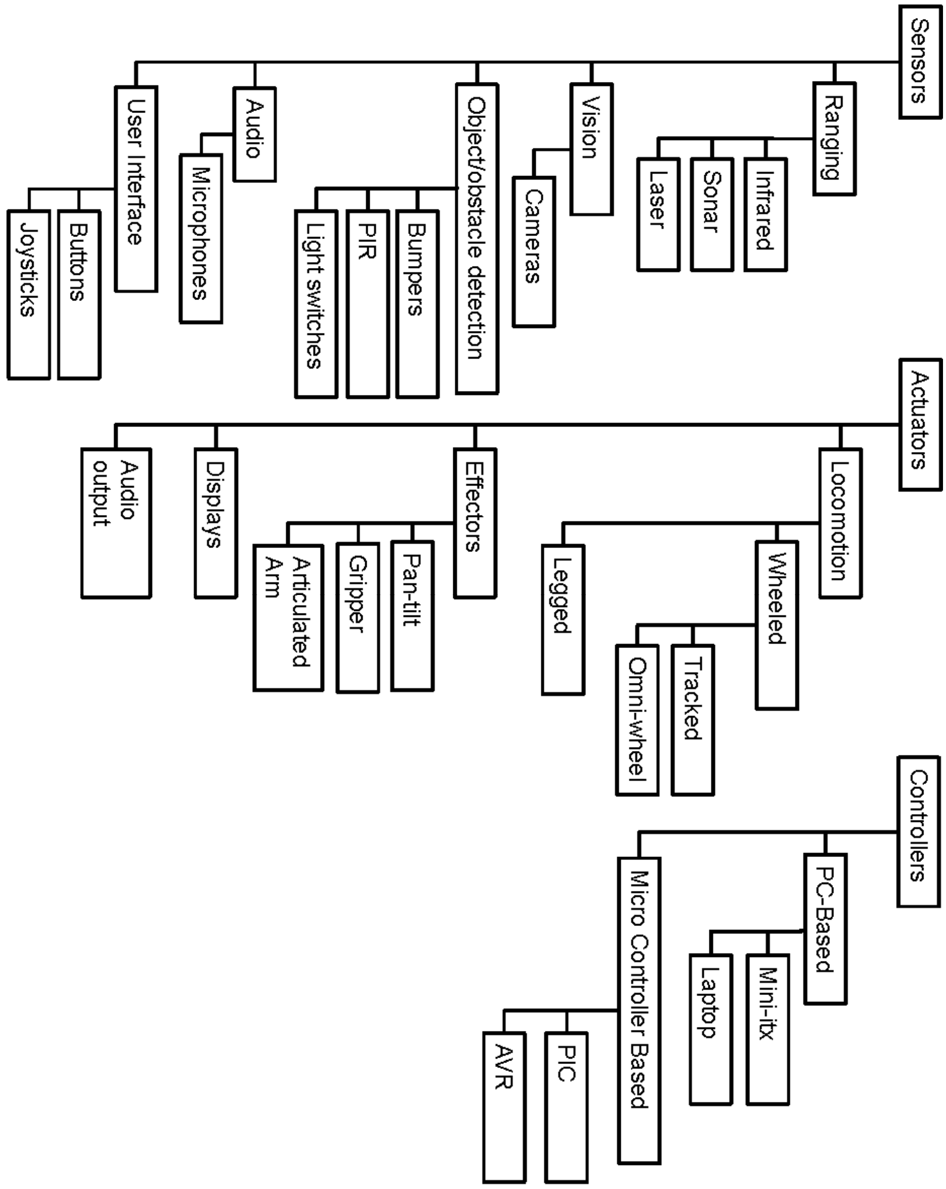


Figure 10: An overview of classification of modules

CHAPTER 4

PROTOTYPE DESIGN

4.1 Module Structure

The conceptual design phase is resulted in a layered, physically modular architecture. For reflecting this layered structure of the robot, it is favorable to use a stackable mechanical construction. As one of the aims of this thesis work is to use off-the-shelf components as much as possible, it is decided to use a general purpose commercial product, or re-purpose a suitable product to constitute a structural base for a unit module.

Among material alternatives like metal, wood or cardboard, polymer based products became an evident choice, because they are easily available in many different geometries, reasonably strong and inexpensive. There are several choices (in forms of plastic boxes, buckets, food containers, cases etc.) with various sizes, strength, durability, therefore selection of the module structure is rather a matter of finding a suitable size that can at least contain the mainboard, which is the biggest component with dimensions of 170x170x40 mm. The other important characteristic is the strength to carry the weight, which is mostly due to the battery. Several alternatives have been inspected in local stores, and two products had been tested. At the end, a storage box made by Hammarplast Company is chosen (Figure 11), mainly because it was more rigid and compact, it provided more usable volume, it was easier to stack and it was also possible to use the lids.

4.2 Locomotion Module

Wheeled locomotion is the type of locomotion chosen for the prototype, as it is the most commonly used type and it is easier and more convenient compared to others. As outlined in the previous chapter, there are several configurations and options for wheel type and placement. As it is the simplest type, differential drive configuration is used (Figure 12).

Figure 11: Storage box from Hammarplast

Accordingly, there are several decisions made. First the position of the motors and wheels should be decided. Maneuverability and traction characteristics of wheels are strongly dependent on this decision. The turning radius of the robot can be minimized by placing the effective wheel axle close to the center of the chassis, which requires using pallets, skid-steering configuration or at least two castor wheels at the front and back of the module.

There were mainly two alternatives available for actuation in locomotion layer. The first one is a DC motor with a planetary gear head, which was extracted from a power screwdriver. Providing a high torque and possibility of easily coupling wheels using the standard hex bit interface, this option also has some disadvantages. It has a poor efficiency, and it requires high currents for operation. In addition, the gearbox is made of plastic and it is not directly coupled with the motor, requiring an extra fixture to keep the package as a whole. And finally, it needed a separate driving circuitry, which needed to be designed or procured. Even though this option was tested for its performance and requirements experimentally, it has been decided not to use it in the prototype due to the disadvantages listed.

The other option is to use modified hobby servo motors for actuation. Their advantages are; they are inexpensive and easy to find, they provide reasonably good performance, and they contain their own driver circuitry. So it is easy to control their speed, by simply providing a pulse width modulated signal. They are originally manufactured for remote controlled hobby vehicles for angle control, so they provide limited amount of rotation. Therefore, some modifications are needed to make them possible to rotate continuously. Also, their output shafts are short, making it difficult to couple with wheels. Under the light of these considerations, modified hobby servo motors had been selected, since they were efficient, and easy to control.

As the local controller, a PIC based BASIC STAMP board is used. This board was used because it was available, together with the modified hobby servo motors. The board uses a built-in USB/RS232 converter, and it basically acts as an interface between the main controller module and the actuators. It is designed in such a way that it only works with external power, which is a 4xAA battery pack in this case.

Final requirement for the locomotion module is wheels. Several alternatives had been reviewed, and it was seen that wheels for roller-skates are easy to find, and inexpensive. After testing a few kinds, an unbranded 70 mm skate wheel is chosen. Wheels are coupled with circular horns that come with hobby servos, and then connected to the actuators.

The basic operation of this module is as follows. The main controller module sends a motion directive, in terms of moving forward or backward, or turning to each side. The local controller receives these commands and generates the required pulse width modulation signals to the actuators for a specified duration. At the end of this action, the local controller gets into listen mode again, waiting for a new command. Until a new command is received, no actuation is produced.

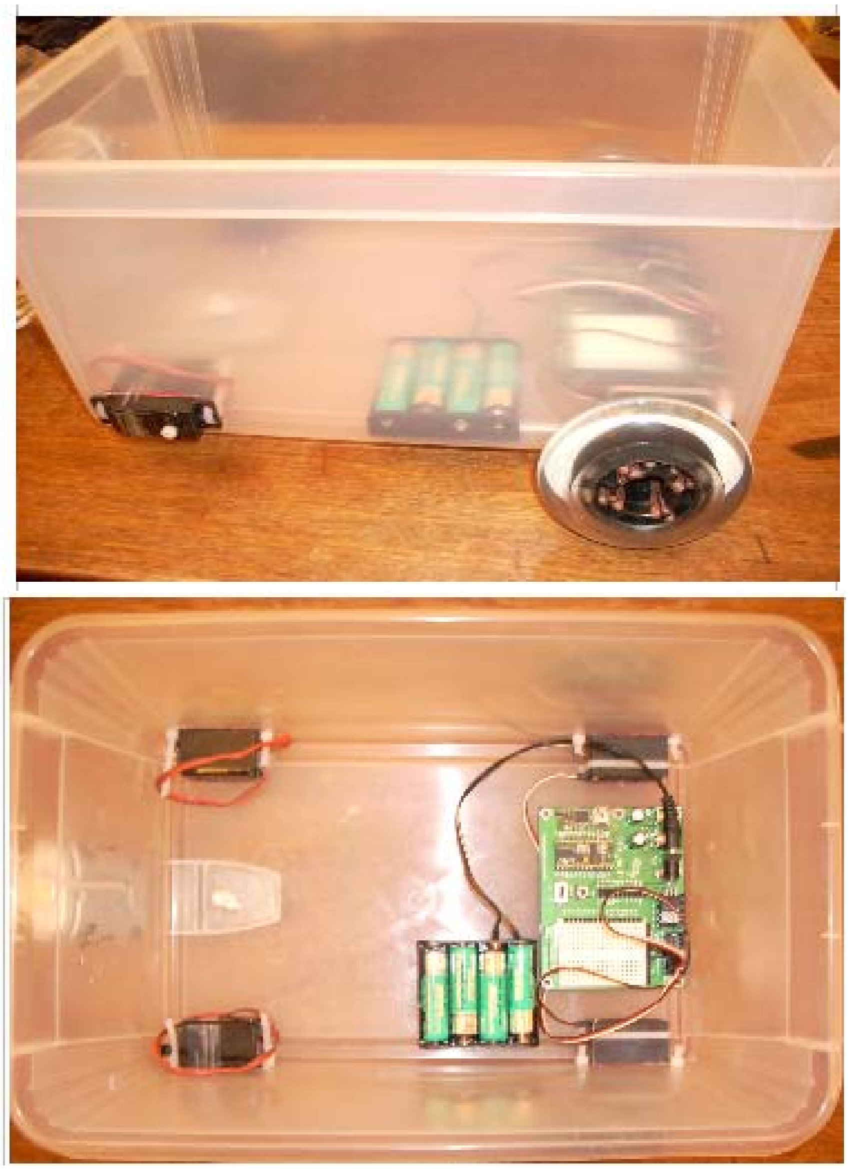


Figure 12: Locomotion Module; side and top views

4.3 Controller Module

This is the central module in the prototype. It basically is a PC, and its peripheral components (Figure 13).

As underlined before, a mini-itx form factor PC mainboard is used as a basis in this module. There are several products using this form factor, having different features. Company is selected. The main advantage of this mainboard is it provides on-board pin headers for most of the interfaces, enabling flexible interfacing with the rest of the components and the modules of the robot.

Naturally, some other components are needed for the mainboard to function. The particular one among these components is the power supply. It is basically a DC-DC voltage converter, which converts unconditioned voltage input in the range of 9-14V, to 3.3V, 5V, 12V and -12V. Therefore, it can be used with conventional batteries, which are subject to voltage drops during discharge. Maximum power output of the power supply is 80W, which is more than twice of the average power consumption of the mainboard.

4.4 Infrared Range Sensor Module

The infrared sensor module is consisted of four sensors and a custom built interface/controller card (Figure 14). As the sensor, GP2D12 model is used, which is produced by Sharp Company. These sensors output an analog signal depending on range data, and their span is 100-800 mm. The sensors are positioned as; two at the front, and one for each side of the module.

The basic operation in this layer is as follows. The local controller receives a range request from another layer. It then fires the sensors, reads their analog input, and converts to an 8 bit variable. Finally, it associates the readings with the sensors, and sends this information to the requesting module.

Figure 13: Main Controller Module Prototype

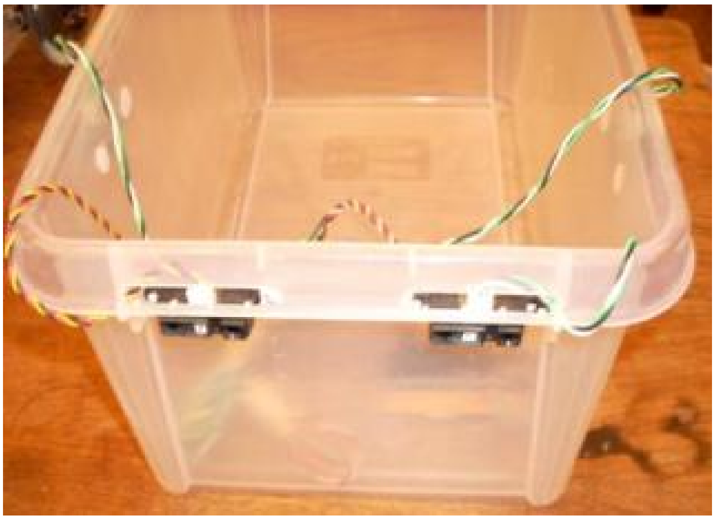


Figure 14: Infrared Range Sensor Module Prototype

4.5 Contact Sensor Module

The contact sensor (bumper) module is consisted of five mechanical switches (Figure 15). Basically, they are unbranded limit switches, which are very commonly used in industry.

These switches are lever actuated, and it is possible to extend these levers and detect contact in a larger distance. They are positioned as; two at the front and one for each side and back. A custom made controller/interface board is used in this module, which connects switches to the main controller.

The basic operation in this layer is as follows. The local controller receives a contact state request from another layer. As the switches are digital sensors, the local controller detects the state of the pins connected to the switches and determines whether there is a contact. Finally, it associates the contact state with the switches, and sends the information to the requesting module.

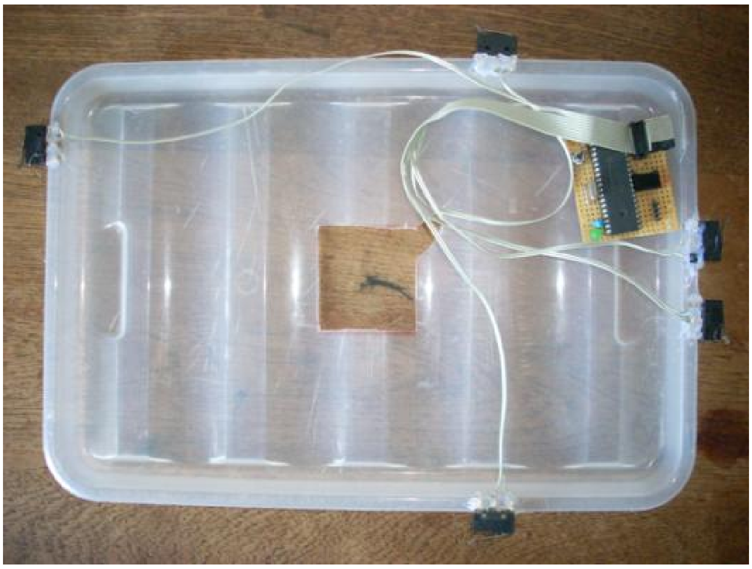


Figure 15: Contact Sensor Module

4.6 Pan-tilt Module

The pan tilt module is consisted of two hobby servo motors, and an Arduino board (Figure 16). These motors are capable of rotating about 180 degrees, and they are attached perpendicularly to each other. Therefore this configuration is capable of scanning a semi-sphere.

The Arduino board acts as an interface, controller and a power source, since this module is USB powered. It is also possible to use an external power source, but since these mini-servo motors do not require more current than the USB port provides, it is currently unnecessary. The principle of operation is very similar to locomotion layer, since the actuators are of same type.

The basic operation in this layer is as follows. The local controller receives position commands for each motor in the form of degrees. Then, it converts the degree data to duration data, which is associated with the width of the pulse signal that are sent to the actuators. Then the motors are synchronously pulsed using hardware PWM of this board, and actuation is produced. Finally, the controller goes into the listen mode, waiting for the next command.

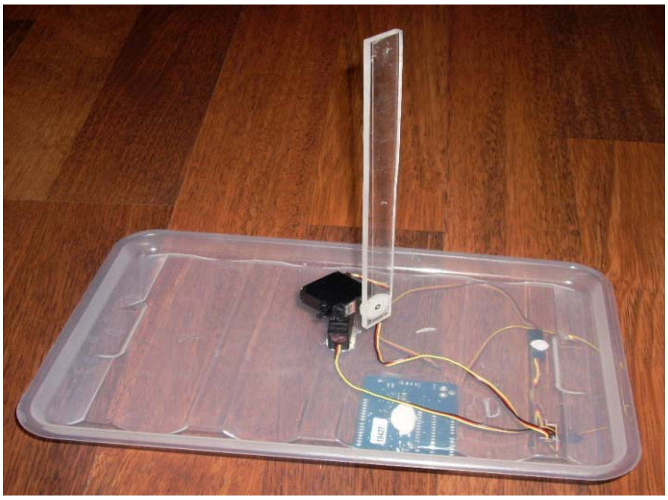


Figure 16: Pan-tilt Module Prototype

4.7 Other Components

In order to complete the functionality of the prototype, some other components are needed. These components are briefly described in the following paragraphs. In addition, full list of components used in the prototype can be found in appendix A.

4.7.1 Wireless Ethernet Adapter

For remote control and monitoring purposes, a USB wireless Ethernet adapter is used, which is produced by Asus Company. In a very compact form, this product enables wireless broadband communication using 801.11a/b/g protocols. Therefore, it is possible to connect to a local network through a wireless interface, or create ad-hoc connections to other local wireless devices, which uses the same protocol.

4.7.2 Webcam

For vision, a standard web camera is used, which is produced by Phillips Company. Using a CMOS sensor, it is capable of capturing 30 frames per second in 640x480 pixels resolution. It connects to the controller module using USB interface, and uses its own proprietary drivers for communication. Therefore, it is only possible to access this device through a high level software interface.

4.7.3 Joystick

In order to enhance the control and show the capabilities of using a PC based controller module, it is decided to use a joystick as an accessory controller unit (Figure 17). The selected device is a cordless gamepad from Logitech Company, which is connected through USB. The gamepad is powered by 2xAA batteries, and it has a range of 8 meters. Principally designed for computer games, this device features two analog and one digital joystick, several buttons and force feedback through a vibration motor. It is possible to access its resources, using freely distributed libraries.

4.7.4 Battery

Due to its availability, a general purpose lead acid battery is used. This battery nominally provides 12V, with a capacity of 7Ah. Measuring 151x65x98 mm, and weighing 2.95 kg, it is housed in the locomotion module, and placed close to the motors.

Figure 17: Wireless Gamepad

4.7.5 Local Interface/Control Cards

It is decided to use USB protocol for interfacing the modules of the prototype. The main reasons for this decision are the availability and convenience, higher speeds, higher number of physical connections and support for bus powered devices.

Therefore, each module is aimed to host a USB controller/interface board that will request sensory information and/or send actuation commands. There are two alternatives in terms of facilitating the USB support in module level. The first one is using a microcontroller that natively supports USB protocol, thus a direct connection between the microcontroller and the main board. A popular choice is the PIC18f4550 microprocessor, which comes in a 40 pin dual in-line package, and features several analog to digital converters and pulse width modulation. It is a suitable choice as it can provide interface to several kinds of sensors, actuators and the mainboard.

Alternative to this method is to convert USB signals to TTL levels, by using a serial converter chip (typically an FTDI 232), and communicate with the microcontroller using RS-232 protocol. It is then possible to use a big variety of microcontrollers that do not support USB natively.

In the prototype, both alternatives had been tried in different variations, in order to see the differences and make comparison between these options. In locomotion module, a BASIC STAMP based board is used, and in the pan-tilt module, an AVR based Arduino board is used, both of which utilize a USB/serial converter. In bumper module and infrared range finder module, custom built boards are used, which accommodate a PIC18f4550 microcontroller. The latter approach is very promising, and it has also been investigated for its possible uses in mechatronics education [30].

4.8 Software Structure

In a mobile robot, intelligence can only be implemented via software. In addition to the mini-itx based main controller module, rest of the modules that utilize local controllers need particular software to operate in an integral fashion.

4.8.1 Operating System

The main controller module is a PC platform. Therefore, it needs to host an operating system as a software platform. As in personal computers, there are mainly three possibilities for the operating system. The first option is the Windows operating system family. Evident options are Windows Vista/XP/XP embedded, and CE. Windows XP is one of the most widely used software platforms in the world. It offers a solid base for development for many engineering fields. But since it is designed as a general platform, it includes several components that are unnecessary for real robotics applications. Windows XP Embedded is an alternative, offering customization of components of the operating system. Windows CE lastly, offers a familiar interface and a platform, mainly for embedded applications. It is possible to run real time applications on Windows CE, but because the programs written for other Windows platforms do not run on CE, it lacks software support.

Second operating system option is OS X, which is developed by Apple Company. It is specifically designed for Apple, but since the introduction of Intel based Mac computers, it is possible to run OS X on commonly used PCs. It usually performs faster than Windows operating systems, but availability of applications and hardware support is less, which makes it difficult to use for beginners. Accordingly, the software support for robotics is fairly limited compared to other alternatives.

The last option is Linux. It is, by its nature, open source and distributed freely. Therefore, there are numerous Linux distributions with different purposes. This is a disadvantage, since there is a lack of standardization among distributions. In addition, computer peripheral manufacturers do not always support Linux operating systems, so there is a significant hardware compatibility issue. None the less, there is a growing community support, and applications for Linux are developed in an evolving manner. Since it is much easier to tailor the components of the system, it is possible to effectively run Linux on computers with limited resources.

For the prototype, Windows XP is selected as the operating system. The main reasons for this choice are its availability and familiarity, and ease of setup and operation. It should also be mentioned that, Windows XP Embedded option has also been investigated and tested; it was not selected since Windows XP already performed well enough on the prototype.

4.8.2 Module Software

In the prototype, four modules (locomotion, infrared range finder, bumper and pan-tilt) utilize a micro controller based platform as an interface to the USB ports of the PC, and as local controllers. Therefore, each module needs its own software to do the specific task. Following of the text gives outlines the needs. In addition, schematics of modules and PIC and Arduino based local controllers are presented in appendix B.

For the locomotion module, the local controller is based on a BASIC STAMP board. This board comes with its own software development environment, which is a BASIC like language. The software is responsible from, receiving direction commands form the main controller and creating PWM signals for servo motors.

For the infrared range finder module, the local controller is a PIC18f4550 microcontroller based board. PIC programs can be developed through a number of different languages and environments, but in this case, again, a BASIC-like language, PICBASIC is used. It has a command set which very similar to BASIC STAMP, since it is designed around a PIC microprocessor. For the range finder module, software is responsible for accepting range data requests, firing sensors, making analog to digital conversions, and transmitting the range data to the USB port of the PC.

The contact sensor module also uses a board same as the one used in the infrared range finder module. The microcontroller in this module simply receives a contact state request, polls the digital lines to mechanical switches, and transmits the contact state information to the main controller through USB.

Finally, the pan-tilt module uses an Arduino board, which is based on Atmel AVR micro controllers. Arduino is an open source project, and it comes with its own development environment [31]. It features a C-like language, and several pre-built libraries for interfacing different hardware. For the pan-tilt module, the servo library is used, which offers an abstract interface to hardware PWM ports of the microcontroller. Software structure is as follows; the main controller sends a set of positions to the Arduino board, which are then converted to PWM signals that runs the hobby servo motors; using the servo library.

In order to interface these modules to the main controller, a document link library is developed in C#. Using this library, it is possible to get or set properties of modules. Therefore, one can develop his own software by using this library, without dealing with specifics of modules.

As each module shares some characteristics, a base module class is generated. This class mainly handles port settings, such as port names, baud rates and existing ports in the system. Sensor classes mainly permits polling of a specific sensor or all sensors in a module at the same time. Locomotion class simply provides directional movement commands. And finally, Pan-tilt class enables setting of pan and/or tilt position in degrees. An overview of the class structure can be seen in Figure 18.

To demonstrate the functionality of modules, a sample program is created. This program enables to user to view the states of modules, control their available properties, and use the vision capabilities; as explained in the previous section. Figure 20 show the graphical user interface of this software.

4.9 Design Finalization

It is possible to say that there is a certain lack of standards in mobile robotics. It is therefore easier to start with the design of the architecture from components, which are present and commonly used, and move on to higher systems. As every part builds upon each other, and is explained with terms described in lower levels, a bottom-up approach is preferred over a top-down approach. It is also apparent that this approach is educational in itself, and better suited to the design of a modular architecture.

In order to finalize this bottom up approach that has been evolving until the exploration of components and configurations for sensors, actuators, controller, communication and power; following decisions are given:

- Centralized approach: The PC based architecture provided standardization in terms of hardware and software. Readily defined physical interfaces and hardware abstraction of the operating system ensures the common usability of the main controller module and its peripherals. Therefore, it is decided to build the system around the PC based main controller module, in a centralized approach.

- Hardware interface: As explained above, the main controller module supports several standard physical connectivity options. In accordance to the decision of centralization, it is decided to use universal asynchronous bus communication. It is very widely used, most of the peripherals already support USB, and it can act as power bus; therefore it is logical to choose USB as the hardware interface.

- Software interface: modules should provide abstract functionality in terms of read/write access to sensors/actuators. A software library is to be provided on the main controller module; as it will be the hardware abstraction layer which is incrementally built as new modules are designed and added. Therefore, each module should be first physically built and then its drivers must be supported.

- Module standardization: each module should house a microcontroller based local controller. These local controllers must be specifically programmed for the task of the module. These controllers are responsible from local processing and abstraction as it also acts as an interface. As the architecture is centralized, these controllers work in USB client mode. It must be possible to manually set the power input either to USB or to external power, as some of the modules might need more power than the USB bus can provide. Each module must be single-purpose so that it deals with only one task even if it has excess resources, in order to establish functional decomposition.

- Module structure: Physical dimensions of the module are set. As it is intended to provide a generic module structure, dimensions are defined in two dimensions. The height of the modules can be set according to the components used in that module. The drawings of the structure are given in the appendix.

- Local controllers: During the prototyping phase, several possibilities had been reviewed and it is decided to recommend the AVR based Arduino platform. Arduino is an open source hardware and software platform; it can be either obtained pre-assembled or its design blue prints can be used for replication. Alternatively, the PIC based controller board design is also given in the appendix.

- Main controller hardware: It is already specified to use a PC based platform. Optimally, a mini-itx mainboard is to be used. In terms of power needs, a local DC-DC converter is used. In terms of storage, it is decided to use a solid state drive. Currently, as they are the most inexpensive alternatives, CF cards are to be used; but if the prices would drop to affordable levels, NAND solid state drives should be used as they are more durable compared to CF cards.

4.10 Minimal and Optimal Set of Modules

As this design is aimed to be used for educational purposes, modularity should meet some relevant requirements. In terms of the architecture, several possibilities for modules have been presented previously. In robotics education, these modules can be used in various levels in different configurations. But since the design aims to target a wide range of audience, it is possible to define a minimal configuration and some additional modules that can be used in most of the introductory levels.

As a must, it is needed to have a main controller module, since it provides the key functionality to the platform. It should be underlined that this module can be used as a standalone PC which can also facilitate the development environment for students. Next, a locomotion module is needed to provide mobility. Since it is the simplest type, it should be a wheeled locomotion module. Finally, a sensor module is needed to provide environmental perception. Minimally, a contact sensor module is needed.

These three modules will be enough to start experimenting with the robotic platform. Even though this configuration seems very limited, robots such as Roomba utilize almost this much functionality to do its task.

Apart from these three modules, it is beneficial to have some additional modules that can be utilized in a wide range of educational activities. A range finder module is very likely to be needed. It can be sonar based, or infrared based depending on the needs and availability. This module will enable the robot to relatively position itself in the environment and contactlessly detect objects. Next, a camera module is would be beneficial. Machine vision applications are increasingly popular so as its education. A layer consisting of USB cameras would be popularly used in robotics education. The number, configuration and functionalities of the webcams might be changed depending on the needs. Finally, a pan-tilt module or a similarly configured gripper module would be used in general robotics education activities.

4.11 Software and Hardware Integration

It is seen that software hardware integration is a difficult and multi dimensional problem. Robots have some essential characteristics or subsystems in terms of sensing, movement, intelligence and energy, as shown previously. Using the modularity approach described, it is intended to hide the variations and complexity of the modules from the logic of the robot.

In order to achieve this integration between hardware and software, it is needed to create an abstraction layer in between the physical components and the software. This abstraction layer therefore allows instructions from higher level computer languages to communicate with lower level components.

In the context of this thesis, this burden of abstraction is shared between local controllers and the main controller. In the module level, main low-level processing occurs. The local controller interacts with actual components, and does the necessary ordering, accessing and computation. Therefore, each local controller is module specific and is developed accordingly.

Yet, each local controller obeys to a certain set of rules to facilitate standardization in the data flow between modules. These controllers support USB based asynchronous communication. In each logical processing cycle of the main controller, a two way communication occurs between the main controller and the connected modules. First, the main controller writes a command to the serial communication line. This can be either a direct command like as in actuators, or an indirect command as in sensors. On the local controller side, these messages are processed and the necessary local commands are executed. And finally, after local processing, the local controllers send feedback messages to the main controller.

In the main controller level, this abstraction is provided in terms of software drivers. First of all, the physical connection of the modules should have a counterpart in the operating system side. Therefore it is needed to set some characteristics of connections. Next, depending on their configurations, modules are able to give feedback in different ways. Each of their different functionality should be supported in terms of a certain set of serial messages to be sent. Therefore, these different functions of the modules should be defined and made it accessible to higher level commands. Each module driver should share a common set of protocols for i.e. communication and different sets of functions to handle module specific tasks. It is therefore natural to bundle these drivers as a library to provide integration support to hardware.

In short, there needed to have two different software development tasks to support the overall modularity concept. First, local control software should be developed to do the actual low level processing, and then a software driver should be developed to support the functionalities of the module in operating system level. Figure 18 shows the overview of this process. Utilizing hardware specific software on microcontrollers and supporting the hardware on the operating system using a ‘middleware’, it is possible to develop custom user programs without hassling with the lower level implementation issues.

Even though further abstraction and integration of the hardware and software is beyond the target of this work, it is possible to forecast possibilities. As mentioned previously, the focus of this work is on hardware modularity, since there have already been various attempts to modularize and standardize robotics software. Providing the hardware modules and their driver support, it is possible to use available middlewares such as Player, Orca or Miro, which provides componentized software support. Therefore, users are able to first install their hardware and then develop their own applications using software components.

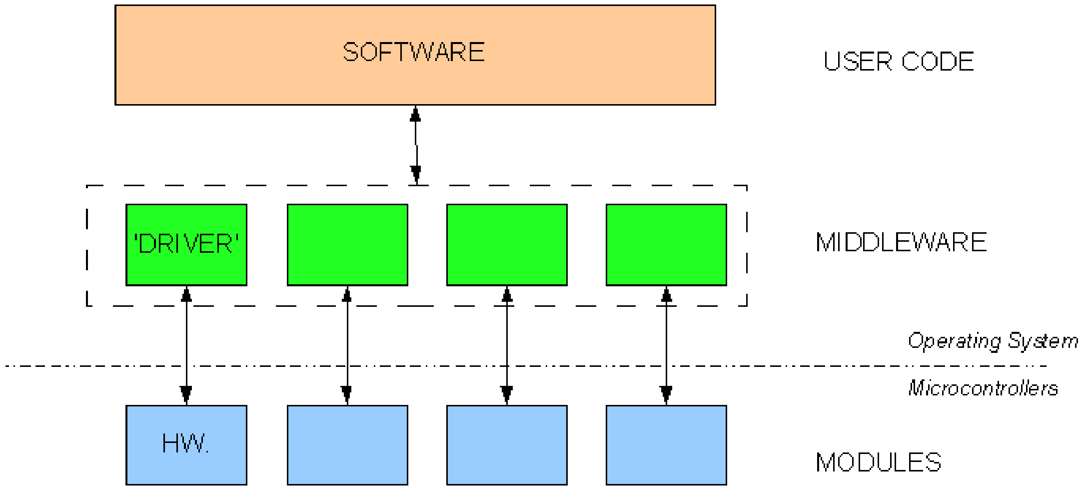


Figure 18: Software structure

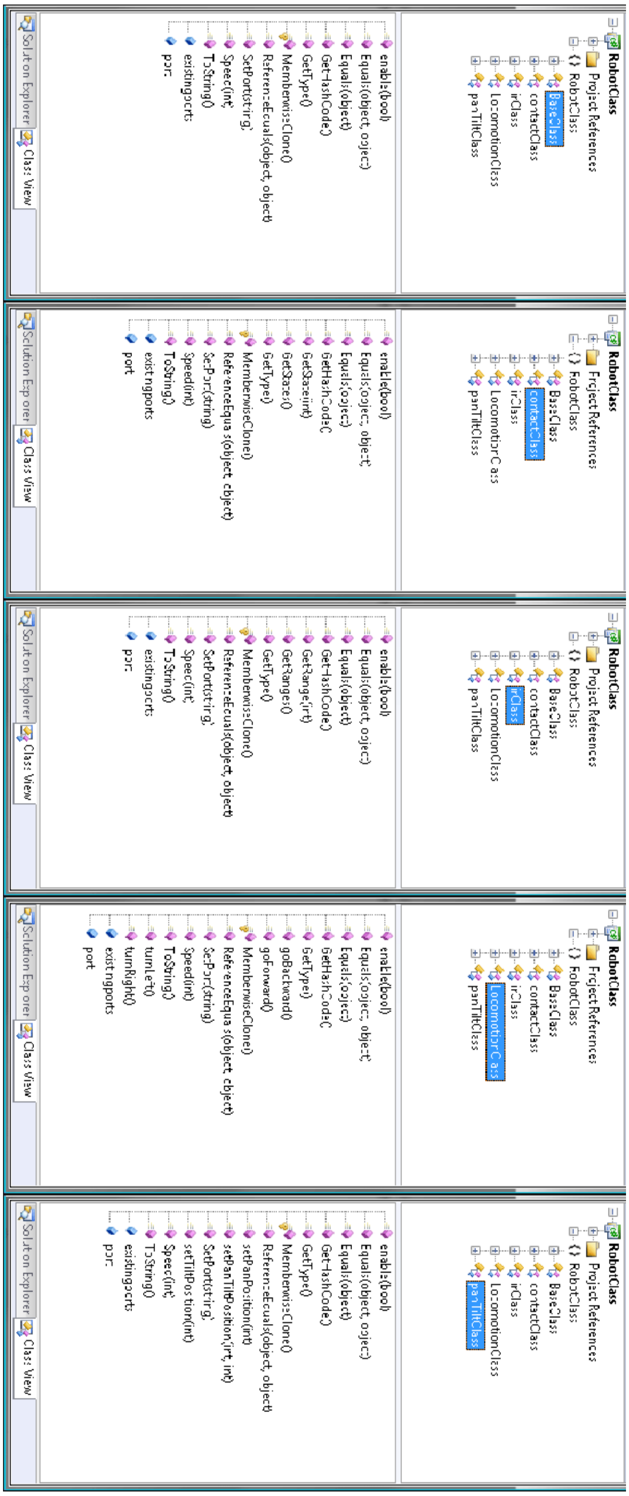


Figure 19: Class view of the module software

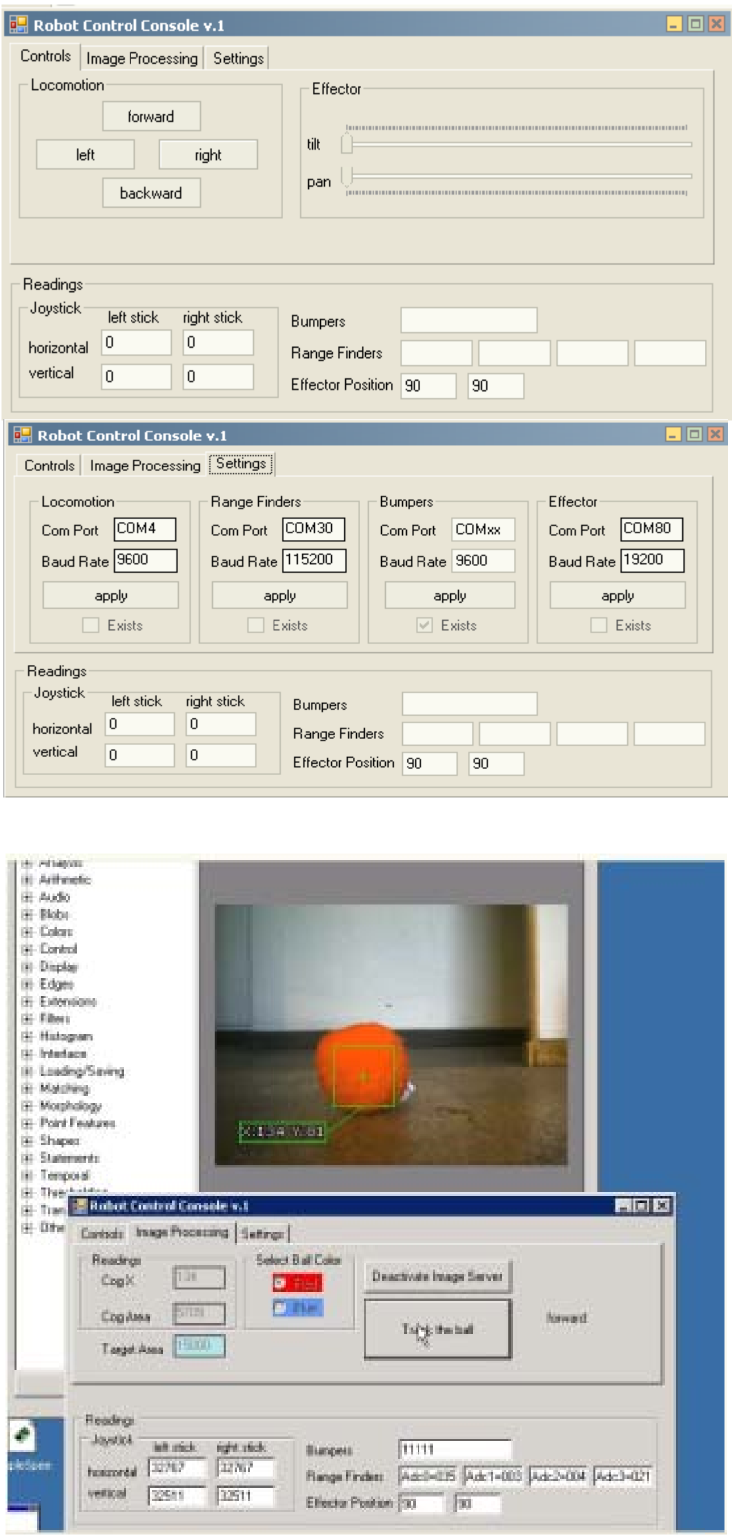


Figure 20: User Interface of the sample software

CHAPTER 5

EXPERIMENTS AND RESULTS

5.1 Software

In order to test the prototype, it was needed to develop software on the main controller module. For that purpose, C# language had been selected, since it is a simple programming language, and it is easy to rapidly develop and deploy applications. As developing environment, freely distributed Microsoft Visual Studio Express Edition is used.

This software is based on three different parts. From the hardware point of view, all of the modules were connected through USB, using virtual serial ports. Therefore, the communication protocol with the modules is a simple text based serial protocol. As a result, setting tab of the program is handling these communication issues; setting ports and speeds of the modules. Secondly, there is a control interface, which gives access to the actuators of the robot. It is possible to give direction commands to the locomotion module, and set the positions for the motors in pan-tilt module. And finally, it is possible to monitor the sensors and actuator states through the monitoring interface. All of the updating of sensory information and sending commands are simply occurs on an interrupt driven basis, by using timers for each respective action.

5.1.1 Vision Software

Visual sensation is done by a USB webcam. As webcams are commonly used devices, manufacturers and operating system developers offer a wide set of high level access tools to their outputs. As it is simply possible to view the live video feed form a webcam, it can be at least used for remote monitoring of the environment.

One of the aims of this thesis work is to use off the shelf components, not only in terms of hardware, but also in terms of software. Image processing and vision applications are very popular in both robotics and other fields of engineering; therefore, there are already a number of vision applications that provides tools for easy development for robotics. Therefore, instead of developing vision software for demonstration from scratch, available options were investigated.

There are several open source or freely distributed vision software, and some notable are; CMVision [32], IMLab [33], and OpenCV [34]. The selected application is Roborealm [35], which is free software, but the source code is not open. Still, compared to the other alternatives, it offers the most extensive processing options, with a very simple interface. Therefore, it is very easy to enhance, filter, adjust or manipulate the vision output coming from the webcam using this software.

In terms of interfacing with other software, Roborealm features a socket based communication, through a locally generated server. Therefore, it is not only possible to interface the Roborealm with the robot control software on-board the main controller module, but also a remote computer. This computer might be more resourceful, and do dedicated or complex image processing and give feedback using a socket based connection.

The program also makes it possible to create a set of filters or actions and save them as configuration files. Therefore, using readily available blob detection and color filters, it is simply possible to make an object tracking application.

The control software interface consequently features vision using Roborealm. Activating the image server basically opens up an instance of Roborealm with a predefined set of filters, runs the server and starts the XML based communication with the server within the application. For the demonstration, a simple object tracking scenario is also developed. Using a red or a blue ball, it is aimed to make the robot find the ball with selected color and approach to it until the image of the ball covers a significant portion of the image, in an autonomous fashion. Once it is asked to find the colored ball, the software runs the

Roborealm which first enhances the image, filters out other colors, filters out the noise and finally computes the center of gravity of the colored object on the image. Then, this data is published through the built-in server of the Roborealm, and accessed by the control software. Depending on at which portion of the image the center of gravity of the colored object resides, control software commands the locomotion module to either turn left or right, or go forward until the area of the colored object is bigger than a specified value.

5.1.2 Joystick Control

Joysticks and gamepads are also highly developed products like webcams. In addition, since they are expected to be used in several different games, their control and accessibility are very important for application developers. As a result, there are several libraries that give easy access to these game controllers in a generic fashion.

Microsoft DirectX [36] is a collection of multimedia application programming interfaces, which focuses on game development. As a result, it exists as a component in almost all of the Windows operating system. Therefore, using DirectX for handling the Logitech cordless Rumble Pad is an apparent choice.

Using the DirectInput library of DirectX, first the gamepad is enumerated and attached to the software. Then, its resources are enumerated, which results in two analog sticks, one directional pad and several buttons. These resources are polled by the software, again, in an interrupt driven fashion, and shown on the software interface.

For demonstration purposes, the left analog stick is associated with the locomotion layer. In each interrupt, it is checked if the stick is moved to a predefined zone and depending on the result, direction commands are sent to the locomotion layer. As the locomotion layer accepts only four directional commands, in case the stick is in a mixed zone (i.e. pointing top left) turning commands precedes the forward and backwards movement commands.

The pan-tilt module can also be controlled by the gamepad. Using the right analog stick, it is possible to move both motors with one degree resolution. Each axis on this stick represents one motor, and it is possible to control these motors at the same time by pointing the stick to in-between directions. Since pan-tilt module accepts degree information, it is possible to assign specific positions to different buttons or button combinations. Using these buttons, it is made possible move motors to their extreme and nominal positions.

5.2 Locomotion Module

Locomotion module has been tested for several configurations. First, it has been prototyped twice with two different module structures. With the first structure, skid-steering was implemented with four individual motors. It did not perform satisfactorily, when the other modules and the battery were added, because the wheelbase was large and there was warping due to the added weight. Therefore, the latter option was tested. This box was less wide and more rigid at the bottom, which resulted in a sturdier locomotion module. Using skid-steering configuration resulted in a rather unsatisfactory performance, mainly due to balancing of the weight, and unpredictable friction differences between two sets of different motors. Finally, a simpler approach was implemented; two wheeled differential drive. Two wheels at the front were removed, and a single point castor was placed as the third contact. Although the resulting module was less maneuverable, it was robust enough to show the proof of concept with locomotion.

5.3 Main Controller Module

The mini-itx board on the main controller was first assembled and used as a stationary PC for testing. This board features a built in processor, a small fan and a large fin structure. In normal use, this module is closed from all sides; therefore it was tested if heating will be a problem. Even though plastics are not very good conductors, the temperature increase was not critical inside the module. It is because the mainboard consumes only less than 35W in average, the hard disk does not generate heat and actually act as a heat sink, and the power supply has a very high efficiency.

The mainboard used in this project is a particular kind of mini-itx series, which provides pin connectors for most of its interfaces. While using it in the robot, only ports used are USB. There are 6 USB ports on board, with two standard USB type-A female connectors. Since the locomotion module, pan-tilt module, camera, joystick and wireless Ethernet adapter uses standard USB connectors, there needed an accessory USB hub to increase the number of the ports. The remaining two modules uses custom made local controller boards, which also provide pin connectors for USB. Therefore, infrared range finder module and bumper module are connected to the mainboard as a daisy chain, using a ribbon cable with 10 pins.

As connecting to several modules and components, and providing power to some of them, the battery operability of this module is also tested. The battery brings 12 volts of direct current, with 7Ah capacity. When the computer was run with the minimum configuration, the battery lasted more than 4 hours. Next, it was tested with the maximum power consuming configuration. In this case, the wireless Ethernet adapter, joystick and the camera are the most power consuming components. In the case of wireless adapter and the transceiver for the wireless joystick, power consumption is continuous, and it is affected by the range and the strength of the signals as well. Camera, on the other hand, uses power only when it is accessed by the system. In addition to these components, pan-tilt module is another power consuming component, which draws current from the USB bus. Since the motors are only lightly loaded, and they only get actuated momentarily, the power consumption related to this module is actually significantly less compared to the previously mentioned components.

In the maximum power consuming configuration, a fully charged battery provided a reliable power for little more than 1.5 hours. After that amount of time, fluctuations started in the USB power, therefore the mainboard shut down some of the USB bus for self protection (Wireless Ethernet adapter is the first component that lost the power). Finally, the mainboard lost the power, where the battery voltage dropped to 11.8 volts. As the battery state of charge in lead-acid batteries is correlated to the battery voltage, 11.8 volts correspond to around % 15 left charges on the battery [28]. This state of charge is close to the critical limit of %10, beyond which this type of lead-acid batteries lose their rechargeability drastically.

5.4 Synopsis of Experiments and Their Results

The prototype built had been tested for its several features. USB connectivity is one of the issues. The mainboard both pin out and standard interfaces to its USB ports, and both were used in the prototype. Software wise, it is seen that using the ports as simulated serial ports enables the usage of simple serial communication protocol. Using a freely distributed programming environment and the developed software library to abstract the functionalities, it is seen that software development for the robot can be very quick. In addition, it is also demonstrated that the functionality of the robot can be highly enhanced using freely available software resources, as it is done for vision and remote control in the prototype.

It is also seen that using off the shelf products as structural elements of the modules is a feasible decision. The plastic boxes used in the project offered inexpensive and easily accessible means for prototyping. Also the wheels used were simply good enough to experiment with the prototype.

Power had been a critical issue for the robot, as the battery operation pretty much determines the operability of the robot. The cost of using a PC platform mainly resulted in increased demand for power, as compared to a microcontroller based robot. Yet, a relatively high capacity lead-acid battery enabled operation for a satisfactory amount of time. It is also seen that this amount is inversely proportional to the added weight to the robot.

CHAPTER 6

DISCUSSION AND CONCLUSION

6.1 Revisiting the Objectives

At the beginning of the thesis, following objectives were given in order to determine the context of this thesis:

- *To design a robotic platform which is inexpensive, easy to use and easy to maintain*: This goal actually sets some fundamental elements of the design. Even though the modular approach brings some redundancy to the system, using off-the-shelf components and open source or free hardware and software, inexpensiveness is achieved. Ease of use is a relative concept, but it is possible to say that the proposed platform meets the criteria set in the definition of ease of use in the beginning. The easiness results from various reasons, such like usage of a PC platform, off-the-shelf components, standardization in terms of software development and availability of software resources for development. Finally maintainability is met through the usage of off-the-shelf components and the modularity imposed. It is fairly easy to replace a defect module with a new one since the components used are highly available and functions of the modules are individualized.

- *To propose modular product architecture as a tool to provide flexibility, easiness and maintainability as claimed*: It is possible to say that this goal and the previous one are highly integrated. As the product design method is set modular, the resulting system obviously reflects the advantages and disadvantages of this approach. The modular approach, as expected, provided the flexibility. In turn, flexibility offered one of the key advantages of using the modular approach for educational use. As outlined previously, it is seen that the need for various different configurations in different levels of education can be satisfied by this property of the design. Easiness and maintainability from the point of modularity is also seen. Stackability and the convenience of the interfaces of the modules make it easy to build robots. And replaceability and reusability of the modules makes it easy to maintain the system.

- *To adopt a PC based platform and investigate the advantages of using this approach:* One of the main promoting features of this system is the usage of the PC platform. This feature brought several advantages to the design. First of all, it brought an integrated development environment. Users are able to work on a very familiar environment and use and make developments. Next, it brought an invaluable standardization. Physical interfaces and operating systems are highly evolved and very easy to use on PC platforms. Consequently, this standardization brought a huge amount of peripheral options. As exemplified in the prototype, devices like webcams or game controllers can be easily adopted to the needs of a robotic task. In accordance, the usage of a PC platform resulted in a relatively inexpensive and more capable system compared to other available educational robots.

- *To use off-the-shelf components whenever possible and reveal the benefits of this approach:* This goal is met in various levels as it has been mentioned several times in the preceding goals. In the prototype, almost all of the components used are inexpensive and highly available. As the aim of this work can be rephrased as providing an open modular architecture for educational use, using such components makes it possible to replicate the design of one developer and further develop it. It is seen that, this way, developing individual modules are also educational activities. If this architecture can reach to a number of users, and the users share their contribution to the modules, it can easily be a community supported open hardware project as well.

- *To show the development potential of such a system in terms of available software libraries and drivers, and hardware peripherals and resources using standardized hardware interfaces:* It is observed that the overall design progress of the system actually showed the development potential of such a platform. Modularity made it very easy to design the system as the design work was much focused and singularly functional. Meeting some simple set of interfacing requirements, it is possible to evolve the modules and master them. The availability of various kinds of peripheral devices also showed the potential in terms of hardware. On the other side, software development potential is also presented. It is shown that open source or free software is much more available compared to the hardware, and the solutions are much diverse. It is shown that there already exist some robotic software platforms which bring modularity to the software development level as well. Providing a library of drivers of the modules, it is pointed to use such a middleware to establish a complete solution for robotics development. Finally, apart from these robotic software platforms, it is shown that using the PC based approach brings up the possibilities of using several software libraries developed for the peripheral devices. As shown in the prototype, it was proven that it was very easy to accomplish autonomy through video processing, using a readily available image processing library.

6.2 Discussion

In this thesis work, it had been tried to outline a design schema of a mobile robot using off-the-shelf components. It was intended to result in a flexible and easy to use system, which can facilitate a basis as a platform for further robotic and mechatronic experiments. Mobile robotics is a very multi-disciplinary field, which also makes it difficult. As it is preceded through robotics education, issues related to software; like machine learning, autonomy and intelligence becomes more dominant. The need for a simple and easy to use platform then becomes apparent.

It was tried to define the taxonomy by functional decomposition, from general to specific;

which resulted in a classified list of modules. Acting as unit-building blocks, these modules made it possible to achieve systems with different purposes and characteristics. During the design of individual modules, two things were kept in mind. First, it was tried to achieve a mechanical easiness, through stackability and module-to-module physical connectivity. Next, it was aimed to provide easy connectivity also in software wise. The modules were provided with local controllers in the form of microcontrollers, which act as interfaces to sensors and actuators, and also to the rest of the system, by pre-processing incoming or outgoing data.

Having the classification of modules, some of the commonly usable modules were designed as the next stage of the design. As a result, represented modules were achieved. It became evident that the abstraction of the functionalities is necessary to achieve better usability. This feature implemented, it is then possible for the main controller to send generic actuation commands and receive generic sensory information.

These modules are enough to start experimenting with the robotic platform. Even though it seems limited, commercially available robot such as Roomba utilize almost only this much functionality to achieve a task that looks complex.

Apart from the modules listed above, it is beneficial to have some additional modules that can be utilized in a wide range of educational activities. A range finder module is very likely to be needed. It can be sonar based, or infrared based depending on the needs and availability. This module will enable the robot to relatively position itself in the environment and contactlessly detect surrounding objects. Next, a camera module would be beneficial since machine vision is becoming increasingly popular. A layer consisting of USB cameras would be used to do image processing and enhance the functionality of the robot greatly. The number, configuration and functionalities of these cameras might be changed depending on the needs. Finally, a pan-tilt, or a similarly configured gripper module might be used as the education further proceeds to the topic environmental modification and autonomous transportation applications.

In order to further study some basic aspects of the conceptual design, a prototype was designed and built. It consisted of a locomotion module, an infrared range finding module, a bumper module, a pan-tilt module, a camera, a gamepad controller, and a mini-itx based main controller module. An important characteristic of the design is the usage of the mini-itx, which provides standard computer functionality. As the computer technology and its peripherals evolved rapidly during last two decades, computers became a central component in various industrial or domestic settings. Robotics consequently affected from this, since many accessory products designed for stationary computers can also be used for robots. The design and the prototype essentially show the simplicity of using this kind of components, such as the camera and the joystick.

Investigating this simplicity led the design to another key concept; connectivity. Common feature of these commercial products mentioned is that, it is possible to connect, and if needed power them using USB. Featuring hot-plugging, it is also possible to disconnect and reconnect USB devices without restarting. And more importantly, the system gains self-awareness, making it capable of knowing its resources. In terms of robotics, these properties become very advantageous, since it is possible to design proprioceptive, self-diagnosing robots.

While designing the modules for the prototype, it was therefore decided to use these advantages of USB connectivity. This decision led to another investigation, where similar functioning microcontrollers were tested and used in different modules. In addition to two ready-made solutions (Basic Stamp, Arduino), it was also decided to custom build PIC18f4550 based boards. The latter choice has already been studied in detail in [30], which also supports the arguments of connectivity and modularity as a case study.

Unlike the alternatives, PIC microcontroller features a true built-in USB functionality. Although it was not implemented in the prototype, this enables the possibility of using it as a generic human interface device. In that case, it is possible to write device drivers for each module and use them as specific devices, rather than emulating serial ports.

Although it has not been stressed on, cost is obviously another factor that affects the design.

Using consumer grade, off-the-shelf components as much as possible, a rather inexpensive design is achieved. As an example, while CMUcam (a camera particularly designed for robotics applications) costs about 180$, a simple webcam costs not more than 30$, providing much better functionality. In addition, whereas special robotics components, such as serial wireless adapters, can be very difficult to obtain; consumer products, such as USB wireless adapters, are much more available. As a result, it is shown that this strategy not only results in a cost effective system, but also provides better maintainability since faulty components can be easily replaced.

6.3 Conclusion

Mobile robotics is an emerging field, which sets new challenges to robotics every day. The breadth and depth of the topics covered significantly affects learning in robotics education. Due to its much applied nature, it is very beneficial to utilize simple and easy to use platforms in robotic education.

With this thesis work, it has been aimed to outline a simple, inexpensive and easy to use mobile robotic platform using off-the-shelf components. The robot system had been handled as a product, and the design approach mainly centered on three key issues; modularity, flexibility and ease of use.

It is shown that by functionally classifying modules, it is possible to obtain standardized modules. It was aimed to abstract modules in as high levels as possible, therefore the system do not deal with the specific module, but the class that module belongs. There already have been some very successful attempts for standardization of robotic software [37], [38]. The intent of this work has been to outline a design proposal which will move the modularization also into the hardware level.

Motivation behind this project was the author's self desire for a decent, inexpensive and customizable robot platform, which is probably shared by many other roboticists. It is shown that this target is achievable by using consumer level, off-the-shelf components. Especially, using a small form factor PC opens up many possibilities, and facilitates a very easy environment for development. It is also noticed that the connectivity capabilities of a PC, particularly USB, very much increases hardware and software flexibility. It is believed that, even the level of autonomy can be increased by exploiting the characteristics of this connectivity and achieving self-aware systems. On this matter, next stage might be envisioned as given the task and environmental conditions, the robot system automatically determines the needed configuration and list the modules to be used in that particular task. Therefore, this work can principally constitute a basis for evolutionary mobile robotics research.

In short, it is believed that the design outline presented in here addresses the issues that had been identified in the problem definition stage, which is presented in the first chapter. The advantages of such a modular design have been shown with the design, and possibility of building a capable and inexpensive robot has been shown with the prototype.

REFERENCES

1. **Anonymous.** Definition of Robot. *Wikipedia.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.wikipedia.org/wiki/robot.

2. **Robotics, International Federation of.** *World Robotics 2007.* Frankfurt : United Nations Publications, 2007.

3. **Baldwin, C. Y. and Clark, K. B.** *Design Rules, Vol. 1: The Power of Modularity.* s.l. : MIT Press, 2000. Massacchusets.

4. *A Modular Architecture for a Multi-Purpose Mobile Robot.* **Gerald S., Gordon F., Arndt M. and Franz W.** Kyoto : 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, 2004.

5. *RABBIT-a modular rapid-prototyping platform for distributedmechatronic systems.* **Zanella M.C., Robrecht M., de Freitas A., Horst F.A., Lehmann T., Gielow R.** Porto Allegre : 14th Symposium on Integrated Circuits and Systems Design, 2001.

6. *A "Brick"-architecture-based mobile under-vehicle inspection system.* **Andreas K., David P., Jin-Choon N., Mongi A., David G., Grant G.** 5, West Yorkshire : Industrial Robot: An International Journal, 2004, Vol. 31.

7. **B., Larson T. and Allen.** Botstack- open architecture. *TedLarson.com.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] http://www.tedlarson.com/robots/botstack/index.htm.

8. **D., Walters.** Oap. *Open Automation Project.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] opa.sourceforge.net.

9. *Task Inference and Distributed Task Management in theCentibots Robotic System.* **Ortiz CL., Vincent R., Morisset B.** Utrecht : Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, 2005.

10. **ActivMedia.** Amigobot. *ActivRobots.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] http://www.amigobot.com.

11. **M-, Woehrer.** Mac Mini Robot. *Oklahoma University.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] cs.ou.edu/mmr.

12. **ActivMedia.** Pioneer Robots. *ActivRobots.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.activmedia.com/p3dx.

13. **K-Team.** Khepera III. *K-Team Corporation.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.k-team.com/.

14. *Demonstrating the Capabilities of MindStorms NXT for the AI Curriculum.* **F., Myles.** Palo Alto : Proceedings of 2007 AAAI Spring Symposium (Robots and Robot Venues: Resources for AI Education Track, 2007.

15. Vex Robotics Design System. *VEX.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] http://www.vexlabs.com/.

16. Parallax - Education. *ParallaxCorporation.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.parallax.com.

17. **Siegwart T., Nourbaksh I.** *Introduction to Autonomous Mobile Robot.* Massachusetts : MIT Press, 2004.

18. **Anonymous.** Definition of Wheel. *Wikipedia.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.wikipedia.org/wiki/wheel.

19. **Thrun S., Burgard W., Fox, D.** *Probablistic Robotics.* Massachusetts : MIT Press, 2005.

20. **Ekahau.** Ekahau Positioning Engine. *Ekahau.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.ekahau.com.

21. *The sonar ring: Obstacle detection for a mobile robot.* **S., Walter.** Raleigh : Proceedings of IEEE International Conference on Robotics and Automation, 1987.

22. *Mobile robot sonar sensing with pseudo-random codes.* **Jorg, K. and Berg, M-.** Leuven : Proceedings of IEEE Conference on Robotics & Automation , 1998.

23. **SICK.** SICK LMS Laser Measruing System. *SICK Industrial Automation.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.sick.com.

24. **HOKUYO.** Hokuyo URG-04Lx Laser Range Finder. *Hokuyo.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.hokuyo-aut.jp/02sensor/07scanner/urg.html.

25. *Service robots – the end of the beginning?* **M., Schofield.** 6, Emerald : Industrial Robot: An International Journal, 1999, Vol. 26.

26. **A., Bulucea.** *Mini-ITX Power Solutions.* Silicon Valley : Ituner Networks Corporation, 2003.

27. **D., Sawicz.** Hobby Servo Fundamentals. *Lecture notes of MAE-412 Microprocessors.* [Lecture notes of MAE-412 Microprocessors]. Princeton : Princeton University, 2006.

28. **I., Buchmann.** *Batteries in a Portable World.* St-Jean-sur-Richelieu : Cadex Inc., 2001.

29. **Anonymous.** Definition of Solid State Drives. *Wikipedia.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.wikipedia.org/wiki/Solid_state_disk.

30. **Ö., Nursal.** *Modular Embedded System Design / Implementation for Mechatronics Education and Research.* Ankara : Master's Thesis, Department of

Mechanical Engineering, METU, 2007.

31. **Banzi M., Cuartielles D., Igoe T.,Martino G., Mellis D.** Arduino: Open-source electronics prototyping platform. *Arduino.* [Online] 1 1, 2005. [Cited: 2 8, 2008.] www.arduino.cc.

32. **J., Bruce.** Cmvision. *Department of Computer Science, CMU.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.cs.cmu.edu/~jbruce/cmvision/.

33. **Anonymous.** ImLab. *Sorceforge-net.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] imlab.sourceforge.net/.

34. **Intel Corporation.** OpenCV. *Intel.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.intel.com/technology/computing/opencv/index.htm.

35. **Roborealm Inc.** Free robotic machine vision software. *RoboRealm.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.roborealm.com.

36. **Anonymous.** Definition of Microsoft DirectX. *Wikipedia.* [Online] 2 8, 2008. [Cited: 2 8, 2008.] www.wikipedia.org/wiki/DirectX.

37. *Most Valuable Player: A Robot Device Server for Distributed Control.* **Gerkey B., Vaughan R. , Støy K., Howard A., and Mataric M.** Wailea : Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001.

38. **B., Gates.** A Robot in Every Home. *Scientific American Magazine.* 1, 2007, Vol. 1, January.

APPENDIX A

Bill of materials of the Prototype

Following is the list of components used in the prototype, and their respective approximate prices as of the date of this document. Please note that all of the software resources used in the prototype are open-source or free of charge.

Locomotion module:

- 2 x 70 mm skate wheels 8$
- 2 x Modified Servo motors 30$
- Basic stamp board 60$
- Battery pack with 4 x AA 2100 mAh batteries 15$

Range finder module:

- 4 x Sharp GP2D12 sensors 60$
- PIC18f4550 based local controller 15$

Contact sensor module:

- 5 x unbranded simulated rolling lever type microswitches 3$
- PIC18f4550 based local controller 15$

Pan-tilt module:

- 2 x mini servo motor 40$
- Arduino controller board 40$
- Servo controller shield 5$

Main controller module:

• Via EPIA 13000 MS mini-itx mainboard**	170$
• 2.5” Fujitsu IDE 30GB & 40-44 pin IDE converter	40$
• Kingston 267 MhZ SoDIMM 256 MB RAM	25$
• PowerStream PST-ITX-2 DC-DC converter**	45$
• Asus Wireless USB-Ethernet adapter	25$

Miscellaneous:

• Philips Webcam	25$
• Logitech Wireless RumblePad II	50$

*Prices are given as VAT included approximate prices as of 2.2008

** These items are discontinued. Their prices are given for substitute products:

- VIA EPIA CN13000G
- PicoPSU-80-WI-32V

APPENDIX B

B.1 Module Schematics

Following figures gives the block diagrams of the modules in the prototype, showing used components and connections.

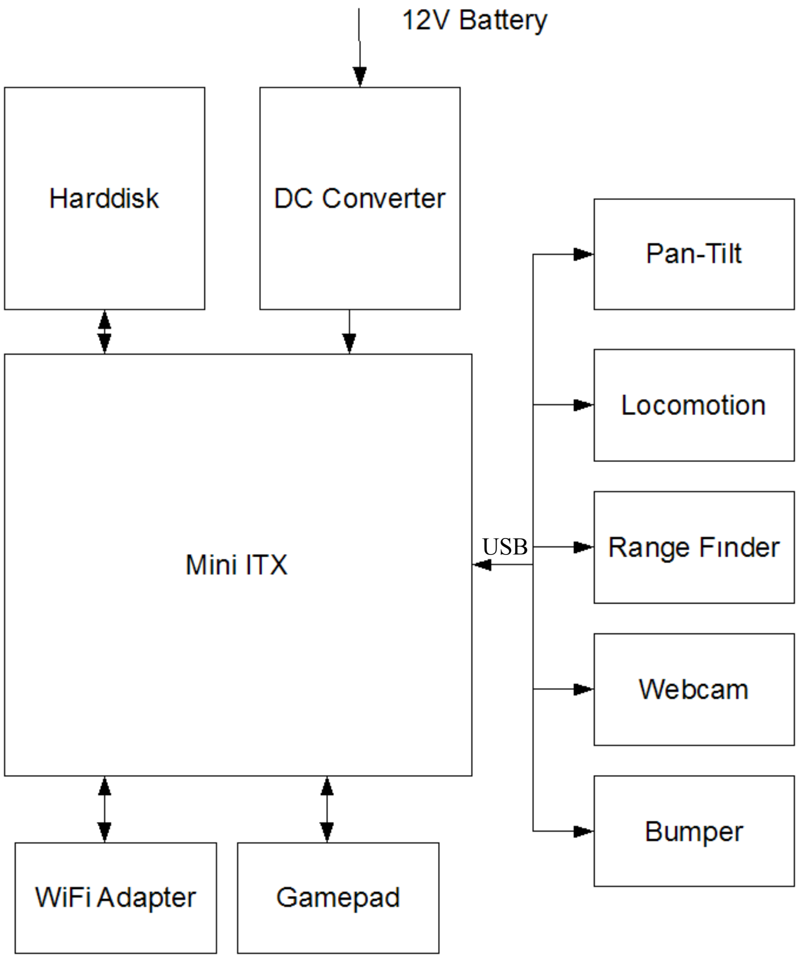


Figure 21: Main controller module

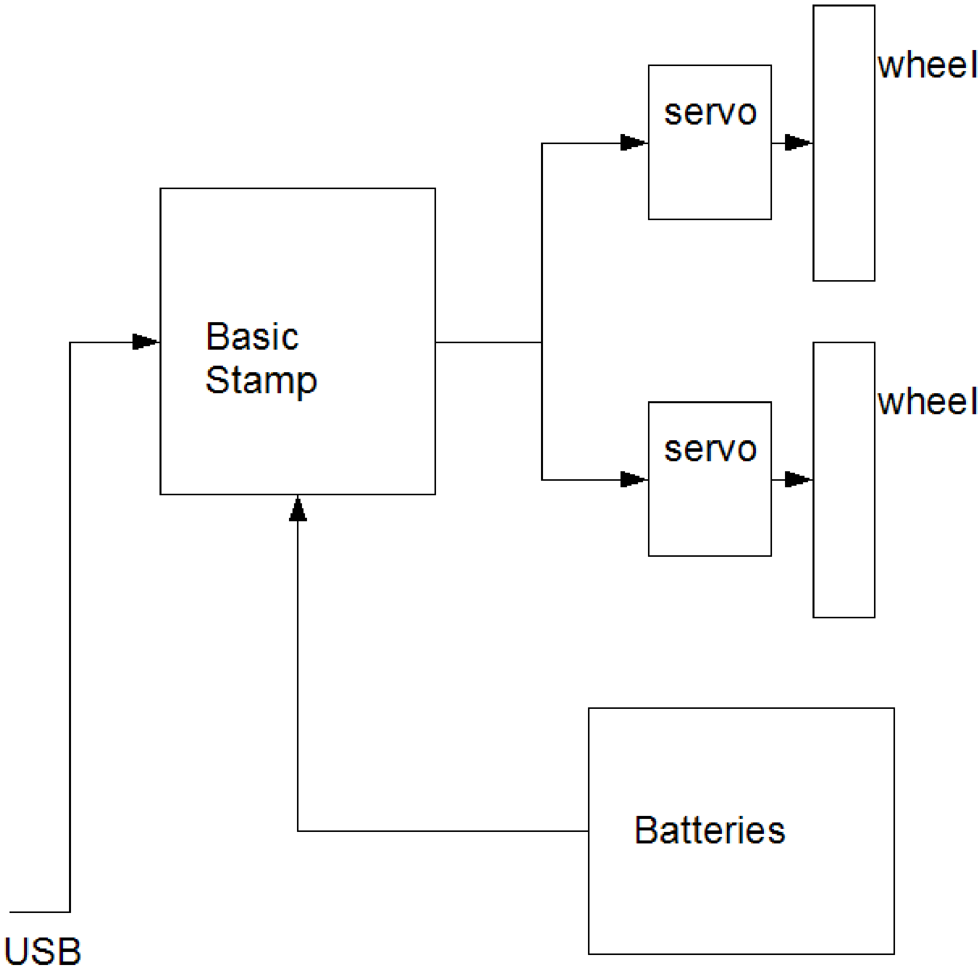


Figure 22: Locomotion Module

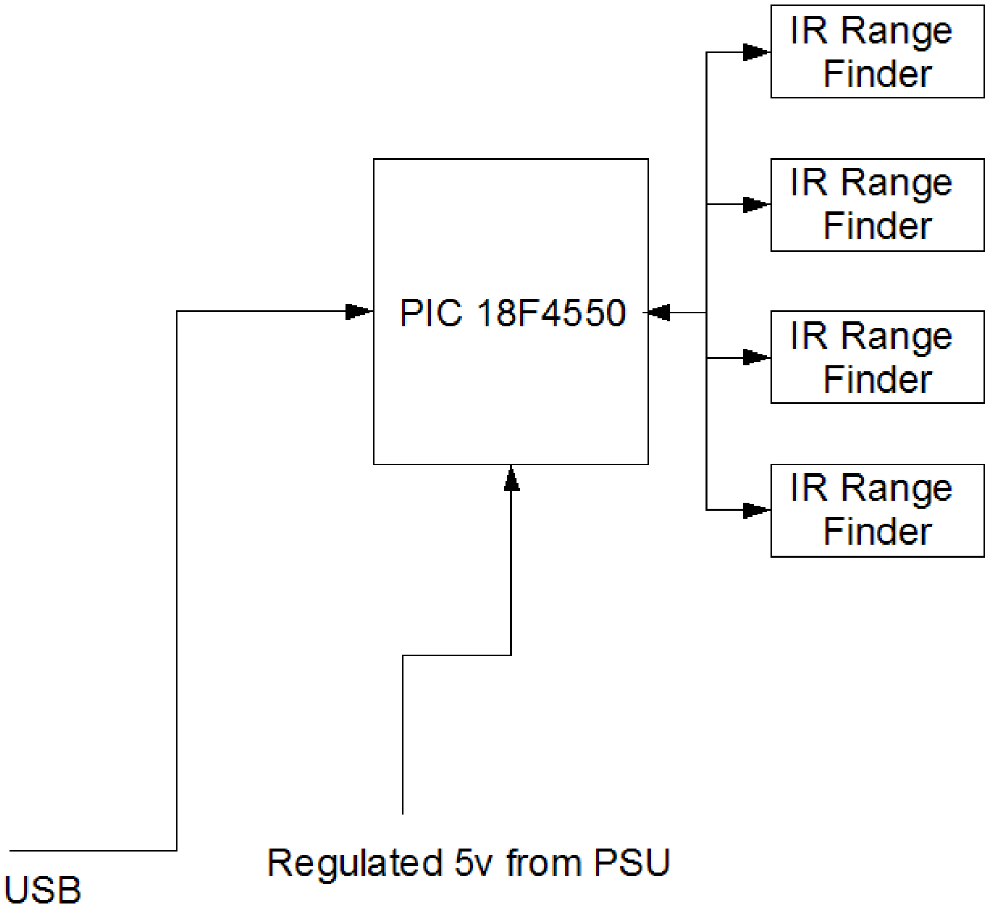


Figure 23: Range Finder Module

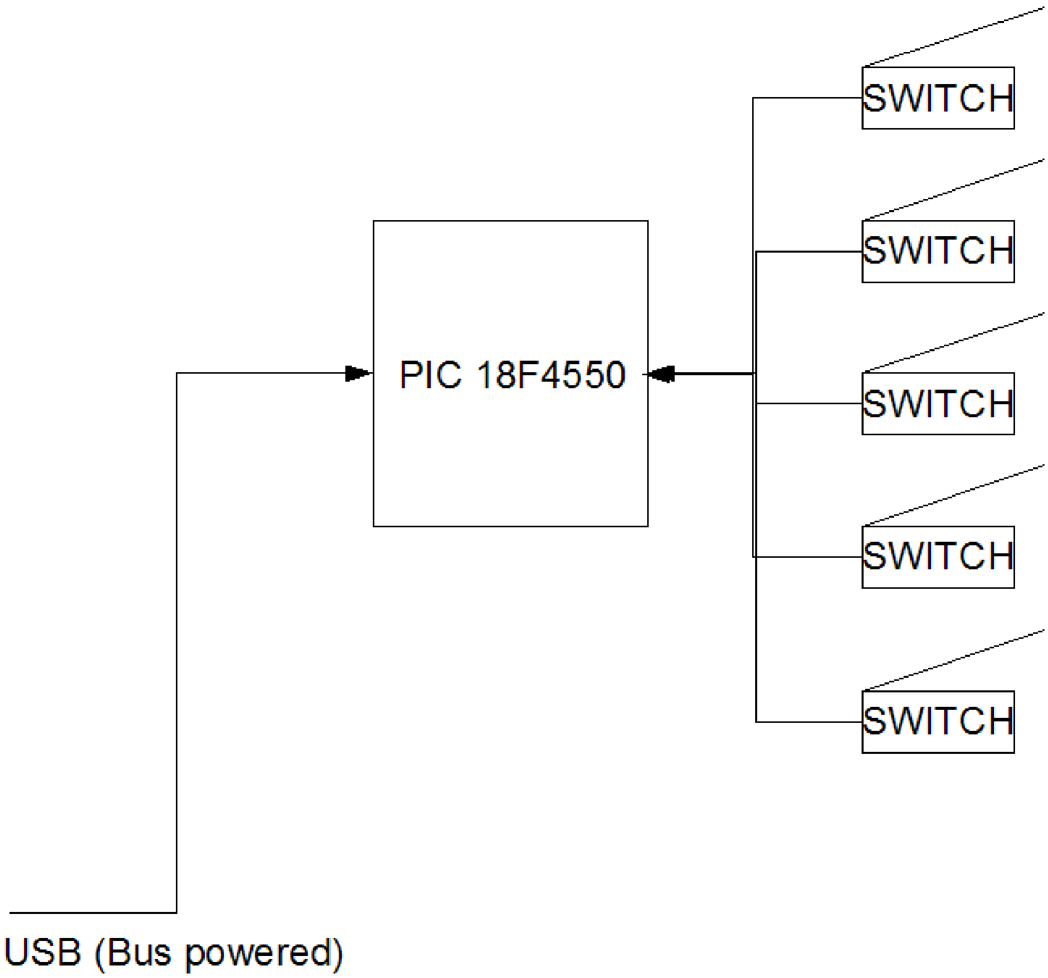


Figure 24: Contact Sensor Module

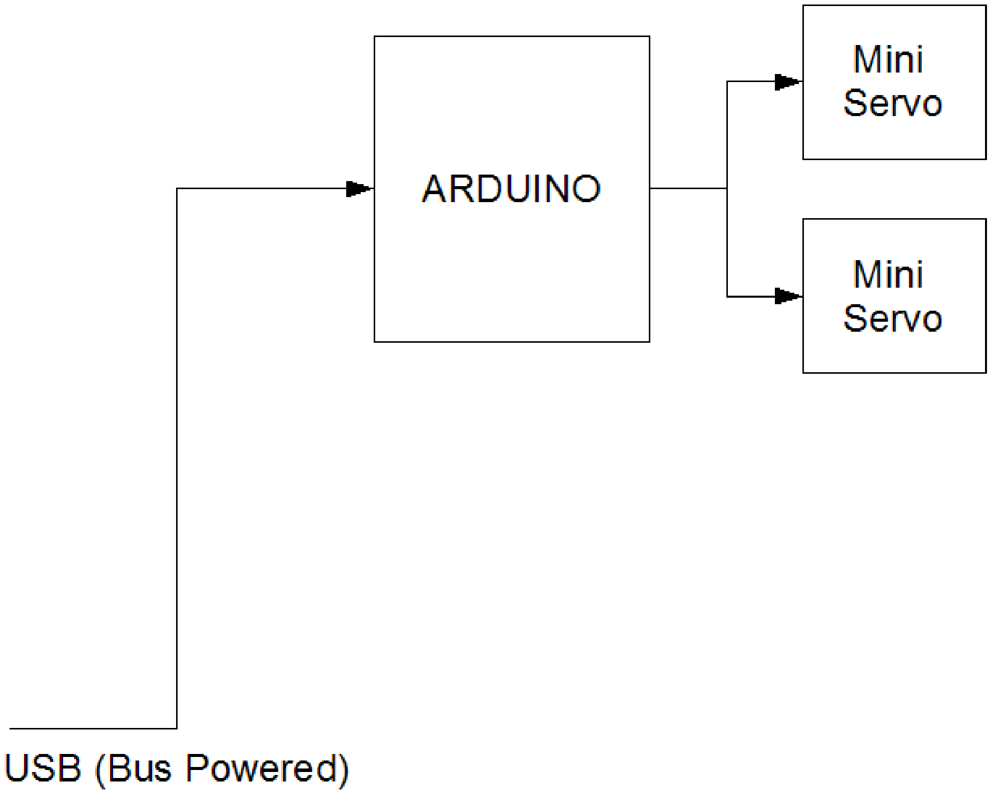


Figure 25: Pan Tilt Module

B.2 Local Controller Schematics

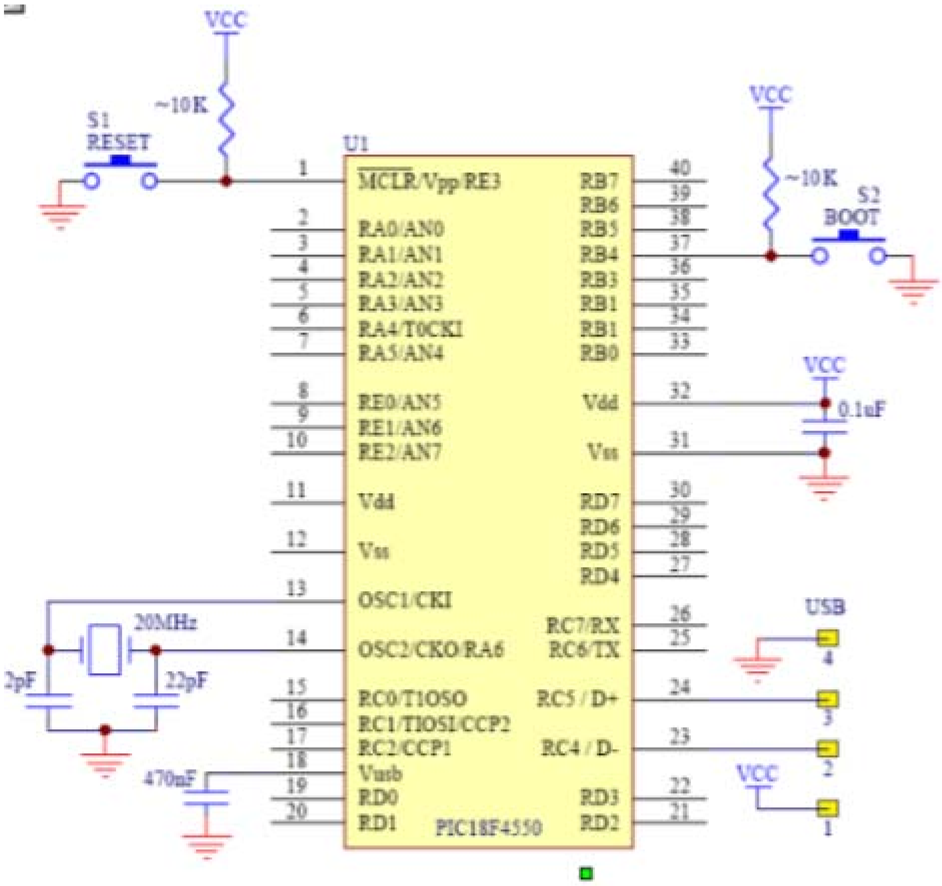


Figure 26: PIC Based Controller

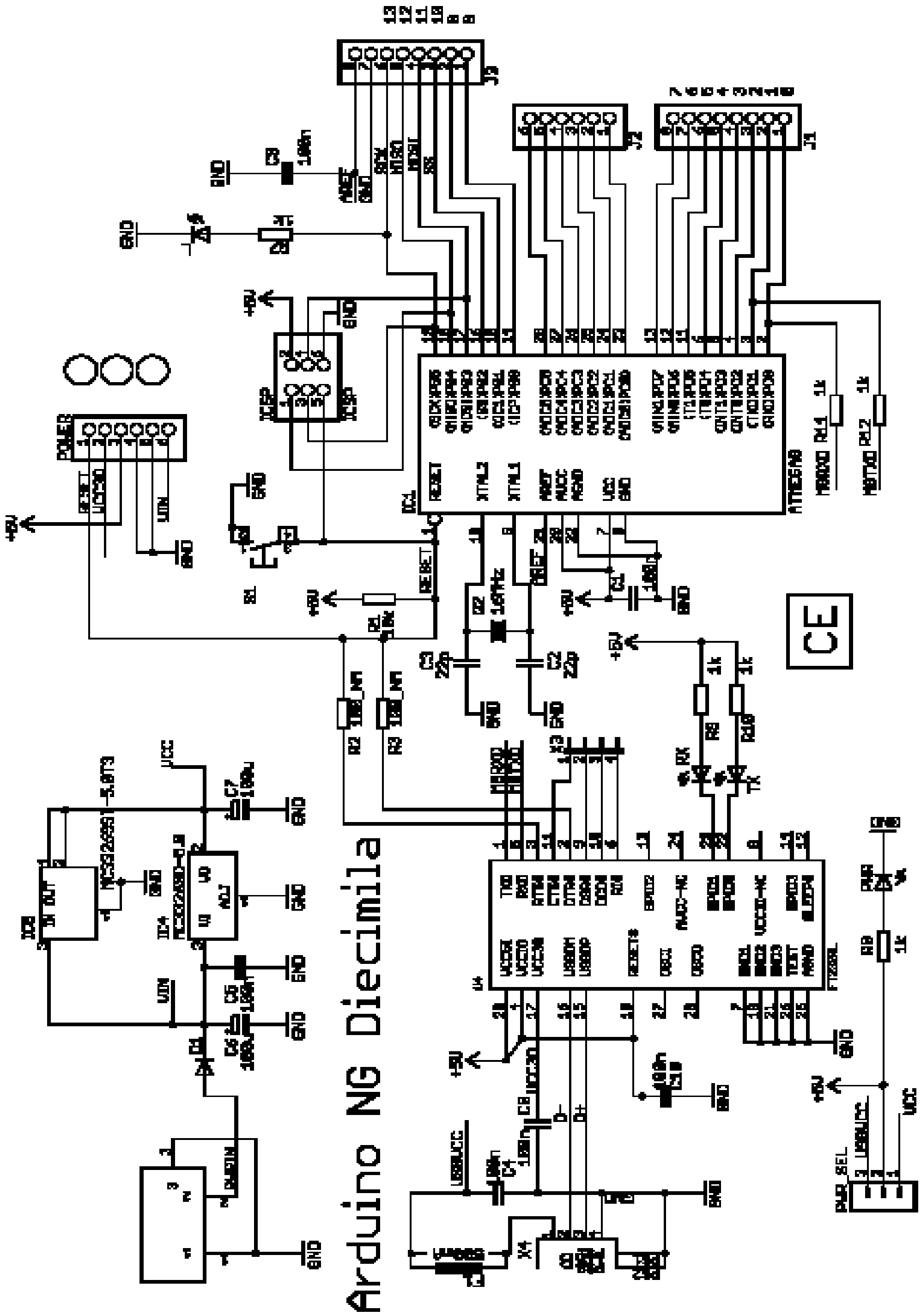


Figure 27: Arduino board schematic