REPUBLIC OF TURKEY
ADNAN MENDERES UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
MECHANICAL ENGINEERING
2015-M.Sc.-013

THE DEVELOPMENT OF FIRE DETECTION ROBOT

Hilmi Saygın SUCUOĞLU

Supervisor:

Prof. Dr. İsmail BÖĞREKCİ

AYDIN

REPUBLIC OF TURKEY

ADNAN MENDERES UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

AYDIN

The thesis with the title of "**The Development Of Fire Detection Robot"** prepared by the H.Saygın SUCUOĞLU, Master Student at the Mechanical Engineering Program at the Department of Mechanical Engineering was accepted by the jury members whose names and titles presented below as a result of thesis defense on January 9th, 2015.

Title, Name Surname	Institution	Signature
President :Prof. Dr. Cengiz ÖZARSLAN	Faculty of Agriculture	
Member :Prof. Dr. İsmail BÖĞREKCİ	Faculty of Engineering	
Member: Assoc.Prof.Dr. Pınar DEMİRCİOĞLU	Faculty of Engineering	

This Master Thesis accepted by the jury members is endorsed by the decision of the Institute Board Members with ………Serial Number and ………………..date.

Prof. Dr. Aydın ÜNAY

Institute Director

REPUBLIC OF TURKEY

ADNAN MENDERES UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

AYDIN

I hereby declare that all information and results reported in this thesis have been obtained by my part as a result of truthful experiments and observations carried out by the scientific methods, and that I referenced appropriately and completely all data, thought, result information which do not belong my part within this study by virtue of scientific ethical codes.

09 / 01 / 2015

Hilmi Saygın SUCUOĞLU

ÖZET

YANGIN ALGILAMA ROBOTUNUN GELİŞTİRİLMESİ

Hilmi Saygın SUCUOĞLU

Yüksek Lisans Tezi, Makine Mühendisliği Anabilim Dalı
Tez Danışmanı: Prof. Dr. İsmail BÖĞREKCİ

2015, 211 sayfa

Bu tez çalışmasının amacı; özellikle endüstriyel alanlarda, erken yangın algılamada kullanılacak bir yangın algılama robotu tasarlamak ve imal etmektir. Bu robot; önceden belirlenen sanal güzergâh üzerinde engel algılama fonksiyonuyla ve yeniden programlanabilir hareket ünitesiyle devriye gezebilecek ve yangın kaynağını tespit edebilmek için ortam taraması yapabilecek şekilde tasarlanmış ve imal edilmiştir. Sistem; hareket planlama ünitesine tanımlanan programlar ile değişken devriye güzergâhlarını takip edebilme yeteneğine sahiptir.

Robotun tasarım ve uygulama süreçleri şu şekildedir; mekanik sistemin tasarımı ve geliştirilmesi, elektronik sistemin tasarımı ve geliştirilmesi ve gerekli yazılımların hazırlanmasıdır. Mekanik sistemin tasarım ve geliştirilme sürecinde; taslak çizimleri, ölçülendirmeler ve üç boyutlu modelleme için bilgisayar destekli tasarım ve katı modelleme programları kullanılmıştır. Robotun taşıyıcı gövdesi; ucuz, sağlam ve kolay işlenebilir malzemeler olan ahşap ve sert plastik köpük kullanılarak imal edilmiştir. Robot sürüş sisteminde diferansiyel metot kullanılmıştır. Yarı otomatik robot dört adet fırçalı doğru akım motoru ile çalışmaktadır. Elektronik sistemin tasarımı ve geliştirilmesi sürecinde; hazır kart almak yerine ihtiyaca uygun elektronik veri kazanım ve kontrol devreleri tasarlanıp üretilmiştir. Bu devrelerin şematik diyagramı ve baskı devresi Proteus elektronik tasarım programı kullanılarak hazırlanmıştır. Bu devreler; motor hareketlerini kontrol etmekte ve dizüstü bilgisayar ile algılama üniteleri arasında bir köprü kurmakta kullanılmıştır. Yazılımların hazırlanma sürecinde; engel algılamada ve güzergâh takibinde kullanılacak akıllı yazılımlar geliştirilmiştir. Ayrıca daha güvenilir yangın algılama sağlamak için; çoklu sensör algılama ve değerlendirme algoritması geliştirilmiştir.

Bu tezin sonucunda; özellikle endüstriyel alanlarda kullanılabilecek, çeşitli fonksiyonlara sahip bir yangın algılama robotu tasarlanıp imal edilmiştir. Yapılan testlerle; sistemin en fazla 100 cm mesafedeki yangını, robot 0,5 m/s hızla ilerlerken tespit edebildiği sonucuna varılmıştır.

Anahtar Kelimeler: Data kazanım ve kontrol, Diferansiyel sürüş metodu, Yangınla mücadele, Çoklu sensör algılama ve değerlendirme, Sanal güzergâh takibi

ABSTRACT

THE DEVELOPMENT OF FIRE DETECTION ROBOT

Hilmi Saygın SUCUOĞLU

M.Sc. Thesis, Department of Mechanical Engineering
Supervisor: Prof. Dr. İsmail BÖĞREKCİ
2015, 211 pages

The aim of this thesis is to design and manufacture a fire detection robot that especially operates in industrial areas for fire inspection and early detection. Robot is designed and implemented to track prescribed paths with obstacle avoidance function through obstacle avoidance and motion planning units and to scan the environment in order to detect fire source using fire detection unit. Robot is able to track patrolling routes using virtual lines that defined to the motion planning unit.

The design and implementation processes of the robot are as follow; the design and the development of mechanical, electronic systems and software. The design and the development of mechanical system; for the sketch drawings, dimensioning and solid state modeling of the robot, computer aided design and solid modelling computer programs were used. The carrier board of the robot is produced using wooden material and rigid plastic foam which are cheap, strong enough and easy to manufacture. Differential steering method is selected for semi-autonomous robot driving system and it is powered by four brushed DC (direct current) motors. The design and the development of electronic system; electronic circuits were designed and produced, instead of buying a commercial card. Both schematic diagrams and circuits of the data acquisition and control circuits are designed using Proteus electronic design program. These circuits are used to control the motion of the motors and establish a data flow between the laptop and the other peripheral sensing components. Software development; intelligent algorithms for obstacle avoidance and path tracking have been developed. A sensor data fusion algorithm for the sensors was also developed to get more reliable fire detection information.

In conclusion; a fire inspection and detection robot with various functions to especially can be used in industrial areas was designed and manufactured. The functions of the robot were tested. It can be concluded that system is able to detect the fire source maximum 100 cm distance away while robot is moving with 0.5 m/s forward speed.

Keywords: Data acquisition and control, Differential steering method, Firefighting, Sensor data fusion, Virtual path tracking.

ACKNOWLEDGEMENTS

I wish to express my gratitude to my supervisor, Prof. Dr. İsmail BÖĞREKCİ who was abundantly helpful and offered invaluable support with his sincerity and belief in me. In addition, I am especially grateful to Assoc. Prof. Dr. Pınar DEMİRCİOĞLU for her useful advices and comments.

I wish to acknowledge the efforts of Fatih AKKOYUN.

Also, I would like to thank my lovely family, for their invaluable efforts when I felt hopeless and weak in solving problems.

Finally, I would like to send my special thanks to my love Duygu for her understanding, help and encouragement and faith in me.

Hilmi Saygın SUCUOĞLU

TABLE OF CONTENT

APPROVAL OF THE THESIS iii
SCIENTIFIC ETHICS v
ÖZET vii
ABSTRACT ix
ACKNOWLEDGEMENTS xi
TABLE OF CONTENT xiii
LIST OF ABBREVIATIONS xvii
LIST OF FIGURES xix
LIST OF TABLES xxv
1. INTRODUCTION 1
1.1. What is Robot? 1
1.2. Problem Description 2
1.3. Motivation 2
1.4. Thesis Objectives 3
1.5. Overview of Thesis 4
1.6. System Overview 5
2. LITERATURE REVIEW 7
2.1. Industrial Fires 7
2.1.1. Big Industrial Fires 7
2.1.2. Fire and Fire Safety Statistics 8
2.2. Early Fire Detection Methods and Devices 11
2.2.1. Optical Flame Detectors 12
2.2.2. Smoke Detectors 15
2.2.3. Heat Detectors 18
2.3. The Evolution of Robotics 20
2.3.1. Robot Manipulators 22
2.3.2. Mobile Robots 28
2.3.3. Biologically Inspired Robots 36

2.4. Related Works (Firefighting Robots) 48
3. MATERIAL AND METHOD.. 67
3.1. Mechanical Part 68
3.1.1. Carrier Board Design and Manufacturing 68
3.1.2. Drawing and Dimensioning of Carrier Board 69
3.1.3. Traction Mechanism 70
3.1.4. Wheels..... 72
3.1.5. Motor-Wheel Connector.. 72
3.1.6. Steering Method 73
3.2. Hardware 77
3.2.1. Power Requirement Calculations 77
3.2.2. Torque Requirement Calculation 80
3.2.3. Motor Selection 81
3.2.4. Energy Supply 82
3.3. Obstacle Avoidance (Ultrasonic Sensors) 83
3.4. Fire Detection Unit 84
3.4.1. Flame Sensor 85
3.4.2. Temperature Sensor... 86
3.4.3. Smoke Sensor 87
3.5. Fire Detection Robot Platform Configuration 89
3.6. Data Acquisition and Control Circuits Design and Implementation 92
3.6.1. DC Motors Control Unit.. 92
3.6.2. Data Acquisition and Communication Unit 95
3.6.3. Servo Motor Control Unit 97
3.6.4. Microcontrollers 97
3.7. Fire Detection Robot System Architecture.. 100
3.8. Fire Detection Robot System Functions.. 103
3.8.1. Obstacle Avoidance.. 103
3.8.2. Operating the Ultrasound Sensors 104

3.8.3. Motion Control 104
3.8.4. Fire Detection 108
4. RESULTS AND DISCUSSIONS 112
4.1. Path Tracking Test with Different Speed Values 112
4.2. Obstacle Avoidance 116
4.2.1. Implementation and Test of the Obstacle Avoidance Structure 116
4.3. Fire Detection Test 118
4.3.1. Detection Test Based on Distances 118
4.3.2. Detection Test on Patrolling 129
4.3.3. Sensor Data Fusion Algorithm Test 132
5. CONCLUSIONS 138
6. RECOMMANDATIONS AND FUTURE WORKS 138
REFERENCES 141
APPENDICES 149
Appendix-1 (Microcontroller Codes) 149
Path Tracking With 0.2 m/s Forward Speed 149
Path Tracking With 0.3 m/s Forward Speed 158
Path Tracking With 0.5 m/s Forward Speed 168
Obstacle Avoidance Algorithm 177
Fire Detection Sensors Data Acquisition 186
Fire Detection Sensors Data Fusion Algorithm 193
Environment Scanning With Servo Motor 202
Appendix-2 (Data Acquisition and Control Circuits) 208
Appendix-3 (Force and Torque Calculator) 209
RESUME 210

LIST OF ABBREVIATIONS

ADC	Analogue to Digital Converter
ADT	Automatic Distinct Telegraph
AFA	Automatic Fire Alarm Company
AGV	Automated Guided Vehicle
Ah	Ampere Hour
CAF	Compressed Air Foam
CAN	Controller Area Network
CCD	Charge Coupled Device
CCTV	Closed Circuit Television
COTS	Commercial off-the Shelf
DAQ	Data Acquisition
DoF	Degrees of Freedom
EEPROM	Electronically Erasable Programmable Read Only Memory
f_r	Rolling Resistance Coefficient
GPS	Global Positioning System
GSM	Global System for Mobile
IC	Integrated Circuit
INS	Inertial Navigation System
IPC	Industrial Computer
IR	Infrared
kΩ	Kilo-ohm
LCD	Liquid Crystal Display
LDR	Light Dependent Resistor
LED	Light Emitting Diode
LiPo	Lithium Polymer
mA	Mili ampere
MCU	Microcontroller Unit
MIT	Massachusetts Institute of Technology
MLR	Multiple Linear Regression
Mosfet	Metal Oxide Semiconductor Field Effect Transistor
ms	Mili Second
mV	Mili Volt
mW	Mili Watt

NBFU	National Board of Fire Underwriters
nm	Nanometer
PD	Proportional Derivative
PID	Proportional Integral Derivative
PLS	Partial Least Square
ppm	Parts Per Million
PSD	Position Sensitive Detector
PWM	Pulse Width Modulation
RF	Radio Frequency
ROM	Read Only Memory
RPM	Rounds Per Minute
SAS	Statistical Analysis Software
SLAM	Simultaneous Localization and Mapping
SPDT Relay	Single Pole Double Throw Relay
SRI	Stanford Research Institute
TTL	Transistor-Transistor Logic
UGV	Unmanned Ground Vehicle
USART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
UV	Ultraviolet
WPAN	Wireless Personal Area Network
ZMP	Zero Moment Point

LIST OF FIGURES

Figure 1.1. Fire detection robot 6
Figure 2.1. Distribution of the number of fires by years 10
Figure 2.2. Distribution of loss of life by years 10
Figure 2.3. Distribution of financial damages by years 11
Figure 2.4. Flame detection distance (Inverse square law) 15
Figure 2.5. Working principle of ionization smoke detector 16
Figure 2.6. Working principle of photo electric smoke detector 16
Figure 2.7. Main components of aspirating smoke detection system 17
Figure 2.8. Working principle of bi-metal thermostat 18
Figure 2.9. Structure of optomechanical heat detector 19
Figure 2.10. Thermistor curve 20
Figure 2.11. Tesla’s robot boat (a) patent drawing (b) structure 22
Figure 2.12. Unimate robot 24
Figure 2.13. Stanford arm robot 24
Figure 2.14. Operational stocks of industrial robots 27
Figure 2.15. Usage areas of industrial robots 27
Figure 2.16. Operational stocks of industrial robots on country basis 28
Figure 2.17. Mobile robot samples (a) aerial rescue robot (b) underwater robot (c) urban robot 28
Figure 2.18. Model of mobile robot’s functional parts 29
Figure 2.19. Grey Walter’s turtle 33
Figure 2.20. Shakey robot 33
Figure 2.21. Stanford Cart 33
Figure 2.22. Genghis Khan robot 34
Figure 2.23. Khepera robot 34
Figure 2.24. Stanley 34
Figure 2.25. Mobile service robots sales for professional usage for 2011-2012 35

Figure 2.26. Mobile service robots sales forecast for professional usage for 2013-2016 35
Figure 2.27. Mobile service robots sales for domestic usage 36
Figure 2.28. The first dynamically stable system 37
Figure 2.29. Gait types (a) statically stable gait (b) dynamically stable gait 38
Figure 2.30. Armored knight and its inner working mechanism 39
Figure 2.31. Mechanical horse 39
Figure 2.32. Steam man 39
Figure 2.33. General Electric quadruped 40
Figure 2.34. Phony Pony 40
Figure 2.35. The Big Muskie 41
Figure 2.36. OSU Hexapod 41
Figure 2.37. Tokyo Institute of Technology robot series (a) PV-II (b) TITAN-III (c) TITAN-XI 41
Figure 2.38. Waseda robot series (a) WABOT-1 (b) WABOT-2 42
Figure 2.39. Honda robot series (a) P2 (b) P3 (c) Asimo 43
Figure 2.40. Two passive walking models of McGeer 44
Figure 2.41. McGeer's straight passive dynamic walker 44
Figure 2.42. CB^2 robot 45
Figure 2.43. Geminoid robot 45
Figure 2.44. Humanoid robots (a) Kismet (b) Hubo (c) Repliee Q2 46
Figure 2.45. Time evolution of the robotics 47
Figure 2.46. LUF60 firefighting robot 48
Figure 2.47. FIREROB firefighting service robot 48
Figure 2.48. FFR-1 firefighting robot 49
Figure 2.49. Firemote firefighting robot 49
Figure 2.50. MVF-5 firefighting robot 50
Figure 2.51. JMX-LT50 firefighting robot 50
Figure 2.52. SACI 2.0 firefighting robot 51

Figure 2.53. ArchiBot-M firefighting robot 51
Figure 2.54. Thermite T2 firefighting robot 52
Figure 2.55. MyBOT 2000 firefighting robot 52
Figure 2.56. Intelligent security robot 53
Figure 2.57. Fire detection tests of the ISR (a) flame detection status (b) gas leak detection status 53
Figure 2.58. Khoon's autonomous firefighting robot 54
Figure 2.59. Chang's fire searching robot 56
Figure 2.60. Kim's fire and operating scenario 58
Figure 2.61. Impact resistance mechanisms of Kim's robot (a) leaf spring (b) side bolster 59
Figure 2.62. Roberto's fire detection robot 62
Figure 2.63. Moving average test result for 10 seconds time period 62
Figure 2.64. Roberto's system fire detection distance test results (a) 10 cm from the flame (b) 1 m from the flame 63
Figure 2.65. Roberto's system (a) temperature sensor test result (b) luminosity sensor test result 63
Figure 2.66. Guo's exploration-rescue robot 64
Figure 2.67. Octavia robot 65
Figure 3.1. Motor-carrier board connector 69
Figure 3.2. Fundamental cases of differential driving method 74
Figure 3.3. Skid steering method for multi-wheeled robot 75
Figure 3.4. Ackermann steering method 76
Figure 3.5. Omni directional platforms (a) three wheels (b) four wheels 76
Figure 3.6. Independent drive 77
Figure 3.7. Resistance forces 78
Figure 3.8. 12 volt 120 rpm DC motor 81
Figure 3.9. LiPo battery unit 82
Figure 3.10.Ultrasonic sensor 83
Figure 3.11. Arduino flame sensor 85

Figure 3.12. LM 35 temperature sensor 86
Figure 3.13. Circuit of temperature sensor 86
Figure 3.14. MQ-9 carbon monoxide and flammable gas sensor 87
Figure 3.15. Circuit of smoke sensor 87
Figure 3.16. Carrier board 88
Figure 3.17. General assembly schematic of the robot 89
Figure 3.18. Disassembly schematic of the fire detection robot design 90
Figure 3.19. SPDT relay schematic diagram 93
Figure 3.20. Working principle of the diode 94
Figure 3.21. Terminals of the Mosfet 94
Figure 3.22. ULN 2003 schematic diagram 95
Figure 3.23. Pin diagram of LM7805 96
Figure 3.24. Pin diagram of the Max 232 97
Figure 3.25. Pin configuration of Atmega 32 99
Figure 3.26. System architecture of fire detection robot 102
Figure 3.27. Rotation direction of the motors (a) Clockwise rotation (b) Counter clockwise rotation 105
Figure 3.28. Speed control with power Mosfet transistor 107
Figure 3.29. PWM method 108
Figure 3.30. Sensor data fusion structure of fire detection unit 109
Figure 3.31. Flowchart of the obstacle detection function 111
Figure 3.32. Flowchart of the fire detection function 111
Figure 4.1. Path tracking test environment 112
Figure 4.2. Path tracking starting at A point 115
Figure 4.3. Left turn at A point 115
Figure 4.4. Path tracking between B and C points 115
Figure 4.5. Obstacle avoidance test environment 117
Figure 4.6. Avoidence of sixth obstacle (80 cm away from the obstacle) 117
Figure 4.7. Avoidance of sixth obstacle (60 cm away from the obstacle) 117

Figure 4.8. Robot configuration with netbook 118
Figure 4.9. Temperature sensor values-50 cm from the candle flame 120
Figure 4.10. Flame sensor values-50 cm from the candle flame..........................120
Figure 4.11. Smoke sensor values-50 cm from candle flame 120
Figure 4.12. Temperature sensor values-75 cm from candle flame 121
Figure 4.13. Flame sensor values-75 cm from candle flame...............................121
Figure 4.14. Smoke sensor values-75 cm from candle flame 121
Figure 4.15. Temperature sensor values-100 cm from candle flame 122
Figure 4.16. Flame sensor values-100 cm from candle flame.............................122
Figure 4.17. Smoke sensor values-100 cm from candle flame 122
Figure 4.18. Temperature sensor values-50 cm from firewood flame 123
Figure 4.19. Flame sensor values-50 cm from firewood flame...........................123
Figure 4.20. Smoke sensor values-50 cm from firewood flame 123
Figure 4.21. Temperature sensor values-75 cm from firewood flame 124
Figure 4.22. Flame sensor values-75 cm from firewood flame...........................124
Figure 4.23. Smoke sensor values-75 cm from firewood flame 124
Figure 4.24. Temperature sensor values-100 cm from firewood flame 125
Figure 4.25. Flame sensor values-100 cm from firewood flame..........................125
Figure 4.26. Smoke sensor values-100 cm from firewood flame 125
Figure 4.27. Temperature detection comparison of candle flame........................126
Figure 4.28. Flame detection comparison of candle flame 126
Figure 4.29. Smoke detection comparison of candle flame 126
Figure 4.30. Temperature detection comparison of firewood flame....................127
Figure 4.31. Flame detection comparison of firewood flame 127
Figure 4.32. Smoke detection comparison of firewood flame 127
Figure 4.33. Temperature detection comparison between 128
Figure 4.34. Flame detection comparison between candle and firewood flame ..128
Figure 4.35. Smoke detection comparison between candle and firewood flame .128
Figure 4.36. Fire detection based on distance test...129

Figure 4.37. Temperature sensor values for candle flame-mobile mode 130
Figure 4.38. Flame sensor values for candle flame-mobile mode....................... 130
Figure 4.39. Smoke sensor values for candle flame-mobile mode...................... 131
Figure 4.40. Temperature sensor values for firewood flame-mobile mode......... 131
Figure 4.41. Flame sensor values for firewood flame-mobile mode.................. 131
Figure 4.42. Smoke sensor values for firewood flame-mobile mode................. 132
Figure 4.43. Mobile fire detection test 132
Figure 4.44. Fire detection algorithm test 134
Figure 7.1. Data acquisition and communication circuit..................................... 208
Figure 7.2. DC motor control circuit.. 208
Figure 7.3. Servo motor control circuit 209
Figure 7.4. Force-Torque Calculator.. 209

LIST OF TABLES

Table 1.1. System specifications 5

Table 2.1. The selected milestones of the industrial robotics 24

Table 2.2. Functional block comparison between mobile robot and human being 30

Table 2.3. Line tracking methodology 55

Table 2.4. Test results of Khoon's robot 55

Table 2.5. Technical specification of Chang's fire searching robot 57

Table 2.6. Hardware specification of the Kim's robot 59

Table 3.1. The design and the dimensions of carrier board 70

Table 3.2. Details of the wheels 72

Table 3.3. Details of motor-wheel connector 73

Table 3.4. Coefficent of rolling resistance 79

Table 3.5. Full range measurement values of LM35 87

Table 3.6. Names and functions of the components 91

Table 3.7. Port and pin configuration of microcontrollers 100

Table 3.8. Direction-speed and motion control 106

Table 4.1. Path tracking test settings and results 113

Table 4.2. Terms and values of fire detection 133

Table 4.3. Results of the fire detection test 134

Table 4.4. Comparison with Khoon's system 135

Table 4.5. Comparison with Xu's system 136

Table 4.6. Comparison with Roberto's system 137

1. INTRODUCTION

1.1. What is Robot?

As strange as it might seem there is no standard definition for a robot. However, there are some characteristics and features can be used for counting a device or a machine as a robot. First of all a robot has to be aware of what is happening in its environment, needs to be able to move and powered by an energy source, if it is necessary a robot has to be smart enough to satisfy the requirement. Robots can be categorized according to their intelligence as follow;

Manual robot is a system that can sense and convey the motion, gets energy from human power. For example, door hinge. When someone wants to open or close the door, it senses the motion from one side then conveys to other side and get the energy from human's arm.

Semi-autonomous system is able to do everything done by a manual robot. In addition, it has to complete tasks defined by a human. Therefore, sensors and control algorithms are employed for this system's intelligence. For example, a remote controlled mobile robot should be able to understand commands and execute the task. Electrical power, batteries, etc. are used for energy consumption.

The most important thing to counting a system as autonomous is decision mechanism. An autonomous robot can do everything done by the others and it makes decision in dynamic environmental conditions as well. For example, an UGV (Unmanned Ground Vehicle) is able to sense, consider and decide.

Robots are used in wide variety of fields (Gupta et al., 2006). Robot manipulator also known as robot arm is used to perform tasks in industry such as welding, painting, palletizing etc. due to its power, rigid body, speed and accuracy. Recently; the usage area of the robots is shifted from the classical industrial manufacturing robot to service robot (Tajiti et al., 2013). Medical robot has invaded the field of medicine. Although this system hasn't taken place of the medical personnel many robotic applications have emerged in medical area such as laboratory robots, surgery and training of surgery etc. Rehabilitation robot also has been used to help people with disabilities. Mobile robot is a system able to conduct tasks in changeable conditions and different places by a platform and locomotive elements. Locomotive system is varied according to operation environment. In the aquatic and aerial environments, propellers, screws and legs are generally used while; wheels, pallets and legs are used in terrestrial environment.

The main purpose of this thesis is to design and produce a fire detection mobile robot and to test its system functions. This robot should be able to operate especially in industrial areas. So it has to have some system functions which can meet the requirements of industrial usage. The system functions of the proposed robot are listed below;

1. Motion planning and patrolling with the prescribed virtual path lines,

2. Obstacle avoidance function,

3. Scanning the environment for early fire detection and providing information.

After implementation of indicated functions, required tests should be applied to system. The required tests are listed below;

1. Virtual path tracking accuracy,

2. Fire detection capability of the system while moving,

3. Maximum and minimum fire detection distances.

1.2. Problem Description

An industrial fire is a type of disaster involving a conflagration that cause to irremediable harms in in industrial settings. Although firefighting is an extremely hard task it is still carried by human operators so fire-fighters put themselves in harm way while trying to rescue the victims and to get the under control the fire before it becomes graver. After starting of a fire it is almost impossible to control it and recover the damaged area. Therefore, the better way for firefighting is to inspect and detect it, before reached the point of no return. For this purpose, some early fire detection devices and methods have been developed but they have many disadvantages such as necessity of fixing of a wall or ceiling, inflexibility and high cost. This leads to necessity of usage of mobile systems also known as fire detection mobile robot.

1.3. Motivation

The relation between the outputs of the thesis study and real life applications is observed. The results are given below;

1. Instead of fixed early fire detection system, using of a mobile fire detection robot can reduce the investment cost of fire protection systems in the industry. To establish a fixed fire detection system; many cameras, sensors and control center are required. This increases the investment cost. The proposed robot can be a prototype for the advanced systems which may take the place of fixed systems.

2. Nowadays, the work safety in working areas especially in industrial areas is a hot topic discussed in everyday by everyone. Industrial fire and protection is

prime issue of the science of work safety. The proposed system can contribute the science of work safety.

3. There are some early fire detection robot prototypes for industrial usage in literature but they operate with prescribed physical lines using optical sensing method. In this method; it is required to establish new tracking lines for every new working environment. With virtual path tracking; this problem can be solved.

These results are motivated us to design and manufacture a fire detection mobile robot.

1.4. Thesis Objectives

The aim of this thesis is to design and implement a fire detection mobile robot that is able to patrol different areas, inspect and detect the fire occurrence. To complete these tasks there are some requirements;

1. System should enable robot to carry different hardware and sensors. The proposed system is semi-autonomous so it needs various hardware and sensors such as batteries, computer, data acquisition and control unit and fire detection sensors. The carrier board of the robot should be robust and modular enough to carry them.

2. System should have an intelligent obstacle avoidance function that robot can operate in working environment without any collision or interruption.

3. Robot should be able to track the virtual path lines.

4. System should have a fire detection unit which can provide reliable and accurate fire information.

To meet these system requirements;

1. Carrier board is manufactured using wooden material and plastic rigid foam which are robust enough to carry the loads, easy to manufacture and light.

2. An obstacle avoidance unit is produced using two ultrasound sensors. In addition to this unit, an intelligent obstacle avoidance algorithm is developed which enable the robot to carry out its patrolling task without any collision or interruption.

3. Instead of physical path line tracking, virtual path line tracking method is developed. The lines of the patrolling path are programmed to robot through DC motors control unit.

4. The fire detection unit is produced with three sensors; smoke, flame and temperature. Each of these sensors has its own false and misses probabilities. To reduce the miss and false alarm ratio and get more reliable results, three sensors are used and a sensor data fusion algorithm is developed.

1.5. Overview of Thesis

The outline of the thesis is as follows;

In the next chapter “Literature Review”;

1. The industrial fires and statistics, early fire detection methods and devices are presented,

2. Evolution of robotics, the related subjects and firefighting robots are described,

In the third chapter “Material and Method”;

1. Design goals and criteria, mechanical design process and manufacturing are presented,

2. Selection criteria and specifications of hardware, details of the obstacle avoidance unit and fire detection unit are explained,

3. Robot platform configuration and components are described,

4. Data acquisition and control circuits design and implementation process is explained,

5. General system architecture and details of functions are presented,

In the fourth chapter “Results and Discussion”;

1. The applied function tests are described,

2. The results of function tests are presented and discussed,

In the fifth chapter “Conclusion”

1. The conclusion is studied.

In the last chapter “Recommendations and Future Works”;

1. Recommendations and possible future works are discussed.

1.6. System Overview

In this section brief information about the presented robot is given. This section does not include the details of the system. The details of the system are described in next chapters.

The technical specifications of the system are given in Table 1.1.

Table 1.1. System specifications

General Specifications	
Total Weight	1.5 kg (without notebook)
Dimensions	400 mm x 300 mm x 200 mm (Length, Width, Height)
Maximum Speed	0.5 m/s
Battery Unit	3 cells LiPo (lithium polymer) battery unit
Motor Operating Voltage	12 V DC
Motor Power	32 Watt
Motor Torque	2.6 Nm
Minimum Fire Detection Distance	10 cm
Maximum Fire Detection Distance	100 cm
Data Acquisition and Control Unit Specifications	
Operating Voltage	5 V DC
Microcontroller Model	Atmega 32
Number of microcontroller	3 pieces

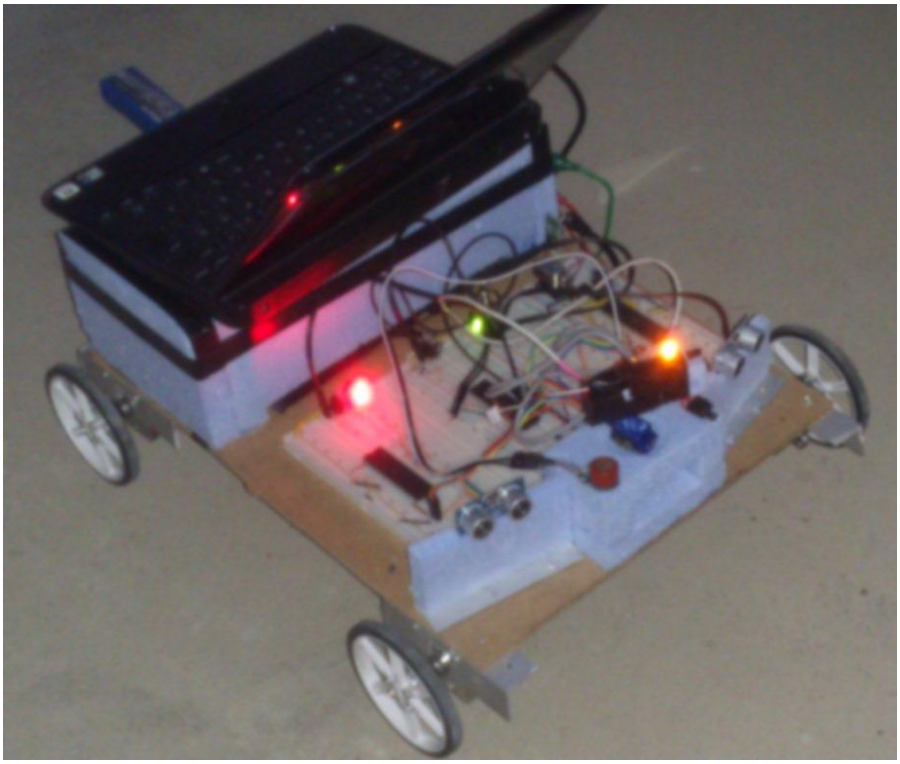


Figure 1.1. Fire detection robot

2. LITERATURE REVIEW

2.1. Industrial Fires

Industrial fire is an industrial disaster which occurs in an industrial place and often but not always occurs with explosions. There are an unlimited number of materials and commodities caused to industrial fire disaster such as chemical products, rubber, plastic, petroleum, wood, clothes, computer parts, furniture etc. Despite knowledge of safety know-hows, industries all over the world still confront frequent explosion and fire hazards (Mishra et al., 2013). These types of fires generally results in environmental pollution and spreading of carcinogenic substances (Haukur et al., 2010). Fire accidents may occur for different reasons ranging from a machine error to personal mistake. The common causes of industrial fires are as follow (Anonymous, 2005);

1. Storing the chemicals in unsuitable conditions
2. Keeping flammable and combustible materials close to ignition sources
3. Keeping incompatible chemicals close to each other
4. Dirty flammable liquids
5. Electrical equipment in bad conditions
6. Overloading electric circuits
7. Keeping hot equipment close to combustible materials
8. Overheating electrical equipment
9. Smoking
10. Not keeping the work area free of thrash, combustible scraps and other debris
11. Keeping machines dusty and oily
12. Not reporting fire hazards

2.1.1. Big Industrial Fires

From history to today many industrial fires have occurred all over the world (Anonymous, 2010 a). Some of them are presented in this part of thesis study;

<u>2012-Dhaka, Bangladesh,</u> Fire disaster occurred in a clothing factory in which 2,000 people work. Consequences of the disaster were death of 117 people,

injuries of 200 and destroying of material for $ 13 million and destruction of factory.

2012-Tuzla, İstanbul, Turkey, Fire started with an erupted blaze in “Kayalar” paint factory on Tuesday Morning. Explosions occurred in factory building and created panic in the area. Fire caused to destruction of 32,000m^2 closed areas and $ 200 million financial damage.

2010-Dhaka, Bangladesh, A garment factory with about 5,000 workers confronted a big fire disaster. 20 people died and thousands of people were injured.

2007-Usak, Turkey, A big fire occurred in Textile & blanket factory “Aran Tekstil”. Fire took almost 4 hours and completely destroyed the 15,000 m^2 closed area.

2005-Bursa, Turkey, The fire started in a mattress factory at about 2:00 a.m. Fire caused many material and financial damage, five workers died in disaster.

2005-Schofield, Wisconsin, USA, One worker died, many people were injured and three buildings were destroyed at a wood product manufacturer in fire. The cost of the disaster exceeded $ 1 million.

2005-Sichuan, China, A chemical plant explosions and fire occurred in a chemical factory. Disaster caused many damages in factory and the entire city. Ten workers lost their life and 21 were injured seriously.

2004- Hiroshima, Japan, The fire started in No.1 plant of “Mazda Motor Corp.” facilities and burned for about 6 hours. There were no injuries or death but shutdown cost was 1,000 vehicles and almost 2.7 billion yen.

2004-Gresik, Indonesia, An explosion followed by a fierce fire in Indonesian petrochemical plant killed two and injured about 50 people. Many buildings burned down in city and the cost of disaster was about $ 75 million.

2000-Enschede, Netherland, The fire occurred in a building in which 900 kg fireworks were stored. It led to an enormous explosion; 27 people died and 947 were injured. A 40 hectare area, 400 homes were destroyed and 1,500 buildings were damaged.

2.1.2. Fire and Fire Safety Statistics

Fire and Fire Safety statistics are organized using some studies about fire information. Fire information can be summarized as below (Brushlinsky et al., 2006);

1. Fire service activities
2. Fire safety products and materials
3. Fire safety science
4. Economic aspects of fire safety
5. Fire education and training
6. Fire safety technologies and related materials
7. Methods of fire suppression

Some statistical fire information (general fires) for Turkey; number of fires, losses of life and financial damages by years are given in Figures 2.1, 2.2 and 2.3 respectively (Bekem et al., 2011).

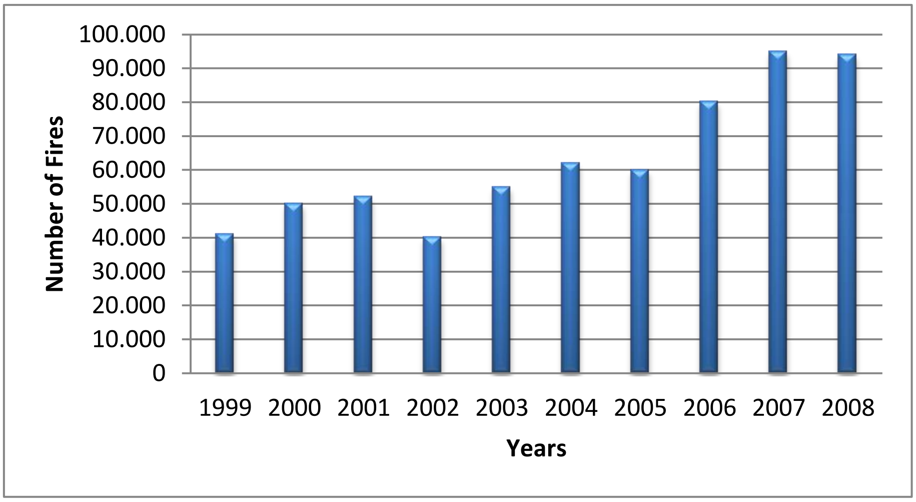


Figure 2.1. Distribution of the number of fires by years

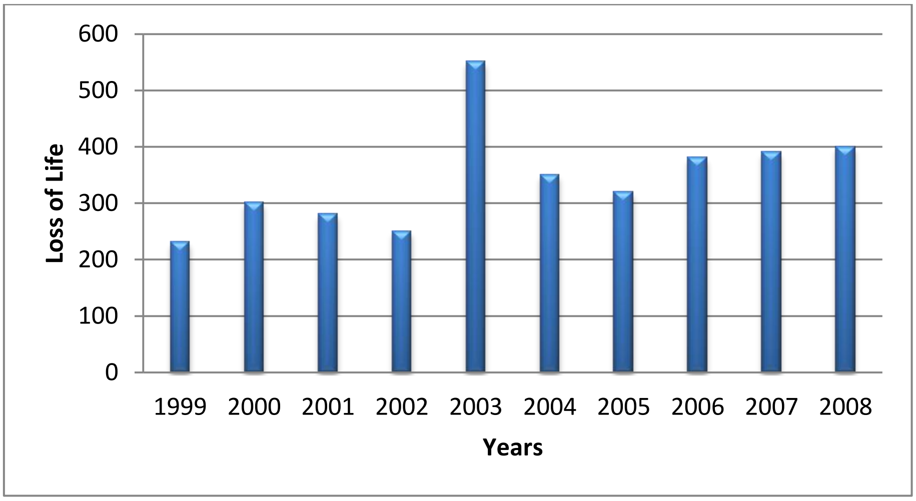


Figure 2.2. Distribution of loss of life by years

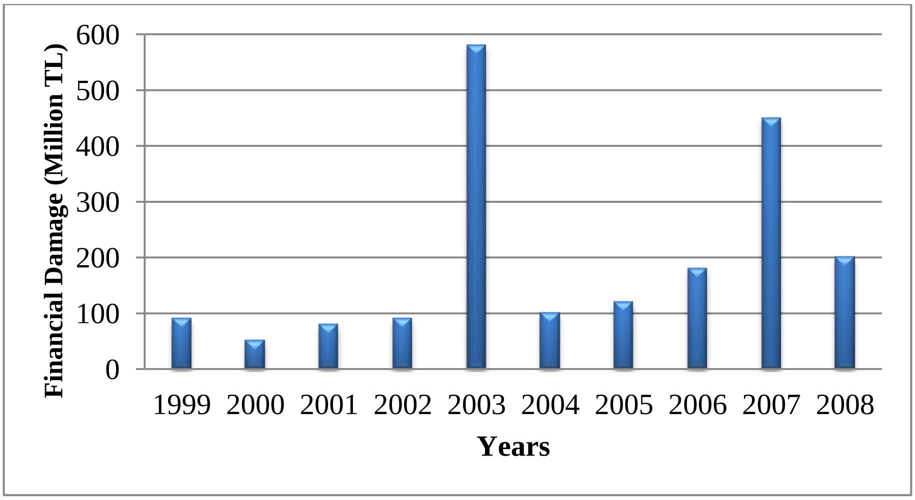


Figure 2.3. Distribution of financial damages by years

2.2. Early Fire Detection Methods and Devices

From the beginning of recorded history, people have learned that early fire detection had positive results in firefighting. In 1847, New York became the first city that used an early fire detection system through telegraph line.

In March 1851, William Channing and Moses G. Farmer installed a fire alarm system based on Samuel Morse's printing register. The system consisted of 40 miles wires to connect the central station to 40 signal boxes, 19 bells in churches, schools and fire engine houses.

The first commercial electrical fire sensor was designed by William B. Watkins. By the early 1870s Watkins had developed remotely monitored fire alarm system using heat detectors and in 1873 established the first private fire alarm company "Boston Automatic Fire Alarm Company" today known as AFA ("Automatic Fire Alarm Company"). The automatic fire alarm telegraph system was operated under any dangerous heat, detected the presence of the fire. The device was mounted the ceiling of every room, office and generally set at 125 °F (52 °C).

In the early 1900s, ADT ("American District Telegraph"), "Holmes Protective", "AFA", AFP ("Automatic Fire Protection") and "Grinnell" contracted support agreements with each other to supply detection, sprinkler systems and equipment, central station monitoring services. As business in the fire alarm and monitoring fields grew, the devices used for detection and protection developed. George Smith patented the first pneumatic detection system "Aero Automatic Fire Alarm" in 1907. NBFU ("National Board of Fire Underwriters") carried out the first heat detector tests in 1956 and published a booklet. "Committee on Fire Research" and

“Fire Research Conference of the Division of Engineering of the National Research Council” proposed a fire research program in 1959 that emphasis on ignition, fire growth and fire spread.

In the mid-1970s the performance of the smoke detectors and their effectiveness in residential environment were tested by “Illinois Institute of Technology Research Center”. The tests were performed in houses located in Indiana and evaluated based on “Escape Time” offered when the detector actuated. According to the experiments smoke detection on every level of home provided sufficient escape time in roughly % 90 of the fire scenarios.

In 1997 a collaborative research program was developed by “The Fire Detection Institute”, “University of Maryland” and “National Research Council” to determine the efficiency of duct smoke detectors in fire alarm system design. Dilution effects, comparative driving forces and smoke ageing effects were investigated (Wayne, 2006).

2.2.1. Optical Flame Detectors

Optical flame detectors have been used as a part of early fire detection systems for many years. An optical flame detector can see fire from long distance up to 65 m away and only 0.01mm^2 in size without need to sense smoke or heat. Detector sees the flame extremely fast due to light speed. Thanks to these features, an optical flame detector is the first option for indoor and outdoor fire inspection and detection applications.

Various types of optical flame detectors are utilized in various spectral bands that are emitted by the fire. Optical flame detectors usually employ several optical sensors that work in specific spectral ranges and these sensors record the incoming radiation at the selected wavelength. Optical flame detectors are categorized as follow (Anonymous, 2009);

1. UV (Single sensor type)
2. IR (Single sensor type)
3. UV/IR (Dual sensor type)
4. IR2 (Dual IR sensor type)
5. IR3 (Triple IR sensor type)
6. CCTV (Image processing type)

To analyze the recorded radiation, some pre-determined methods are used. These are;

1. Flickering frequency analysis
2. Threshold energy signal comparison
3. Mathematical ratios and correlation between various signals
4. Comparator techniques (and-gate techniques)
5. Correlation to memorized spectral analysis

Several of these techniques are employed by the modern optical flame detectors in order to provide accurate and reliable information.

2.2.1.1. UV Flame Detection (0.1-0.4 µm wavelength)

The UV (Ultraviolet) spectral signature of some flames in the range between 100 and 400 nm has a pattern that can be easily recognized over the background radiation. UV detectors can detect the flames at high speed (3-4 milliseconds) based on this technology due to high energy UV radiation emitted by fires. However, this discernible UV radiation can interfere with atmospheric materials such as smoke, smog and vapors or random UV radiation sources such as arc welding, lighting, X-Rays, solar radiation. These can cause the false alarms in outdoor applications. Therefore, UV detectors are mostly employed for indoor applications.

2.2.1.2. IR Flame Detection (0.75-1,000 µm wavelength)

IR (Infrared) sensors can easily recognize the spectral pattern that emitted by flame temperature and production of fire hot gases. However, in a fire scenario the flames are not only source of IR radiation. Oven, halogen lamp and furnaces also emit IR radiation that coincides with flame in IR radiation wavelengths. In order to handle this problem various parameter analysis and mathematical techniques are employed. The most commons are flickering analysis and narrow band IR threshold signals filters between 4.1 and 4.6 µm wavelengths. Most single IR detectors are based on pyro electric sensors with 4.4 µm optical filter and a low frequency 1-10 Hz electronic band pass filter. These systems are mostly used for indoor applications.

2.2.1.3. UV/IR Flame Detection

The UV sensor easily activated by alarm stimuli such as arc welding, X-rays and solar spikes, although it is a good fire detector by itself. An IR sensing channel, working at 2.7 µm or 4.1-4.6µm spectral ranges is added to UV detector to prevent false alarms. This system is considered reliable for most mid-range applications. However, this technology has its disadvantages and limitations since each type of fire has its own characteristic. For example; while a hydrogen flame generates a lot of UV radiation with very little IR, a coal fire generates little UV and high amount of IR radiation. Therefore, the dual UV/IR detector must combine signals and compare these signals to distinguish the other sources caused false alarms.

2.2.1.4. IR/IR Flame Detection

In dual IR flame detection technology, two narrow spectral ranges in the near IR spectral band are used to eliminate the false alarms. Hydrocarbon flames emit energy of a continuous nature in near IR between 0.9 and 3.0 µm, and peak at the 4.3-4.5µm caused by a hot CO_2 fire product; these features are the heart of the most dual IR detectors. Two narrow bands 0.9 and 4.3 µm or combination of short wavelength 0.8-1.1µm and long wavelength 14-25µm IR channels are employed at common dual IR flame detectors. Some of the IR detectors contain a channel in 4.7-16µm IR band for background detection.

2.2.1.5. Triple IR Flame Detection

Combination of three IR sensors is used in triple IR flame detection system. One of these sensors is responsible for the CO_2 flame emission spectral band and the others are employed for specially selected spectral bands, where black body emitters and background radiation are interfering. This system employs some algorithms for analyzing radiation intensity, ratios, correlations, threshold values and flickering signals that obtained from three sensors.

2.2.1.6. CCTV Flame Detection

CCTV (Closed circuit television) flame detection is a system that consists of triple IR flame detector and a color video camera. In this system color video camera can give option to user to investigate the monitored area, identify the fire's source location and help to select the best response to the situation.

2.2.1.7. Detection Distance

Sensitivity and range are related to fire size. The detectable fire size varies according to the inverse square law. If detection distance is doubled, % 25 of the flame lights can reach the detector. Conversely; system needs four times larger area of fire for the same response time. For example, if a standard detector with capable of detecting 1 m^2 fire at 10 m distance is located at 20 m distance from fire source; the required minimum fire size will be 4 m^2. The flame detection distance according to the inverse square law (Anonymous, 2009) is shown in Figure 2.4.

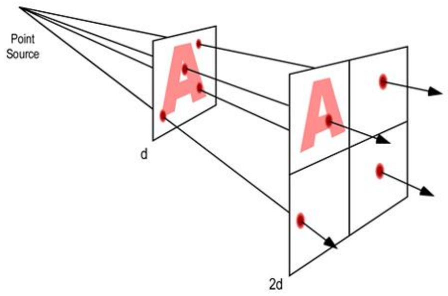


Figure 2.4. Flame detection distance (Inverse square law)

2.2.2. Smoke Detectors

Smoke detectors are known as the most common method of fire inspection and detection to save the life all over the world. There are five types of smoke detectors. Photo-electric and Ionization are the most common types. The others especially used for special applications; (optical) beam, aspirating and video smoke detectors (Porteous, 2011).

These systems are designed to regulate the air flow with a detector and eliminate the foreign materials and insects to reduce the false alarm rate and improve the detector performance.

There are some factors that affect the selection of suitable system for application;

1. Fuel speed of growth
2. Flame and type of smoke produced

For example; while ionization smoke detectors are good at detection of fast flaming fires associated with invisible smoke, photo-electric smoke detectors respond well to slow smoldering fires associated with visible smoke.

2.2.2.1. Ionization Smoke Detector

Ionization smoke detector is the earliest approach for smoke detection that first developed by Swiss physicist Walter Jaeger in 1930. The working principle of the ionization smoke detector can be explained as follow;

Under normal circumstances the air in chamber is ionized by Americium a radioactive element. This causes a free and equal electron flow between two adjacent electrodes. If smoke particles enter the chamber, normal flow of electrons is interrupted and this situation causes alarm activation. In Figure 2.5 this phenomenon is explained (Anonymous, 2010 c).

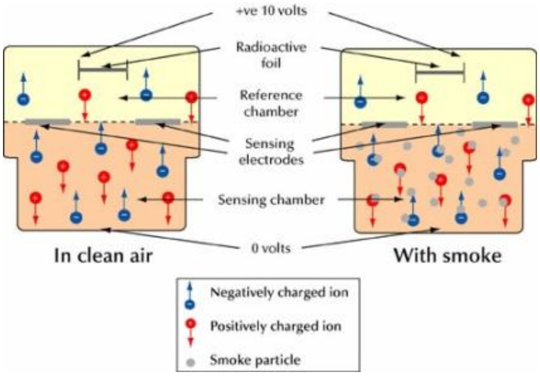


Figure 2.5. Working principle of ionization smoke detector

2.2.2.2. Photo-Electric Smoke Detector

As it can be understood from name a photo electric sensor is an optical detector that contains a transmitter and receiver. The transmitter and receiver are mounted inside a black chamber with suitable offset distance. In normal circumstances, transmitter emits focused light beam. This light is absorbed by wall of the chamber and receiver receives no light.

When visible smoke particles enter inside of chamber, the light emitted by transmitter is scattered in all directions. Receiver detects this light and activates the alarm. In Figure 2.6 working principle of photo electric sensor is explained (Anonymous, 2010 c).

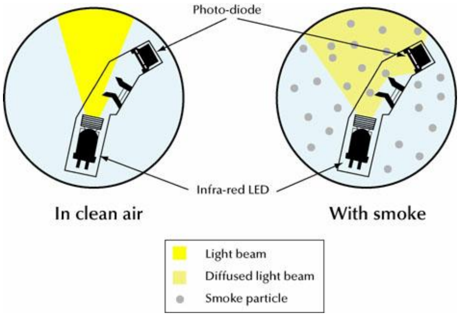


Figure 2.6. Working principle of photo electric smoke detector

2.2.2.3. Optical Beam Smoke Detector

Optical beam smoke detector system employs a combination of focused light transmitter, light receiver and a retro reflective surface (prism) to return beam to the receiver.

A beam smoke detector works on the principal of obscuration. If the light beam reduced or interrupted by smoke particles system is activated and an alarm state occurs. Beam detectors are generally used for long distance applications up to 100 meters. Though it seems as advantages, beam detector requires a straight and uninterrupted direct line of sight. These systems are generally used in hangers, warehouses etc. where the multiple smoke detection methods are impractical.

2.2.2.4. Aspirating Smoke Detector

The working principle of aspirating smoke detector is the same with photo-electric smoke detector. If smoke enters the sensing chamber, smoke particles scatter the light then light is detected by sensitive receiver. These systems are generally used in applications that very early smoke detection is required. Figure 2.7 shows the main components and working principle of aspirating smoke detector.

Aspirating smoke detection systems consist of four main components (Porteous, 2011);

1. A network of pipes with one or more holes to inhale sample smoke
2. A calibrated aspirator
3. A particulate filter
4. A calibrated smoke sensing device

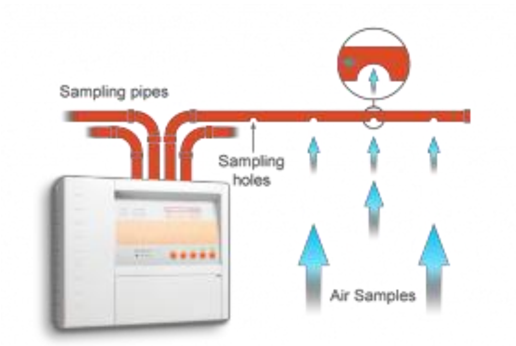


Figure 2.7. Main components of aspirating smoke detection system

2.2.3. Heat Detectors

There are two methods to detect fire from the presence of the heat;

1. When the ambient temperature increases sufficiently to predetermined temperature level, system operates and alarm is activated (Fixed temperature heat detector).

2. When the ambient temperature increases over time equal to or greater than the rate of change, detector operates and activates the alarm system (Rate of rise heat detector).

2.2.3.1. Electromechanical Detector

Electromechanical heat detectors operate due to a combination of mechanical movements and creating an electrical circuit. There are four fundamental types of electromechanical fixed-temperature heat detector (Porteous, 2011);

1. Bi-metal thermostat operates with one end fixed into position and the other end free to move depending on the change in its temperature. When the ambient temperature increases, bi-metal strip creates an electrical circuit that activates the alarm system. Figure 2.8 shows the working principle of bi-metal thermostat.

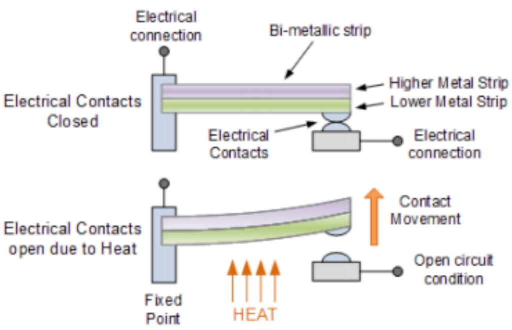


Figure 2.8. Working principle of bi-metal thermostat

2. The most common type of fixed temperature electromechanical detector is a fusible link including a eutectic alloy. Eutectic alloy is the mixture of metals whose melting point is lower than the individual metals. When the ambient temperature increases alloy's temperature, alloy changes the state from solid to liquid. This change releases a spring held under the pressure and activates the alarm system.

3. The third one contains a long heat sensitive wire lined between two points. One point is fixed and the other is draped on a wheel. A weight at the end of

the wire, maintains the tension of it. When the ambient temperature increases, wire expands and weight drops to a point which creates an electrical circuit and activates the alarm.

4. The fourth type consists of a twisted pair of electrical conductors and the heat sensitive insulators that separate these conductors from each other. When insulators exposed the heat, they change physical state from hard solid to molten that enable conductors to create an electrical circuit and activates the alarm.

2.2.3.2. Optomechanical (Linear) Detector

An optomechanical detector is advanced and modern variation of the electromechanical heat detector. The system contains one or more fiber optic cables separated by heat sensitive insulator. A focused light signal is passed through the cable. When exposed to the heat, the heat sensitive insulator changes state from solid to molten. This change causes the degradation of the light signal. The signal changing is monitored by a device and activates the alarm. Figure 2.9 shows the structure of optomechanical detector (Anonymous, 2008).

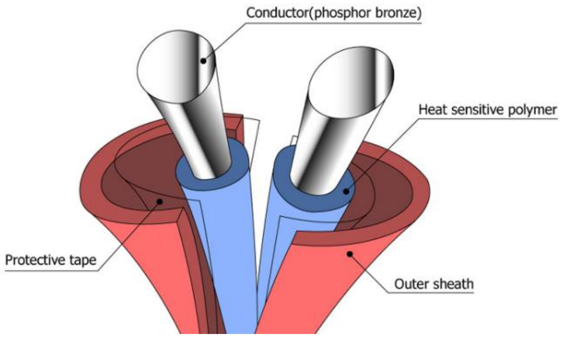


Figure 2.9. Structure of optomechanical heat detector

2.2.3.3. Electropneumatic Detector

Electropneumatic detector is a rate of heat rise detector containing a vented chamber which includes a diaphragm. This diaphragm moves due to pressure changes according to rate of change of ambient temperature. If ambient temperature changes faster than calibrated, the diaphragm moves sufficiently to create an electrical circuit that activates the alarm system. The main advantage of this system is that it can operate at a range of temperature as it has capability to respond the rate of change in temperature.

2.2.3.4. Electronics (Thermistor) Detector

A thermistor is a type of resistor that its resistance changes substantially according to temperature. The resistance of the thermistor is high in low temperature ambient with little electric current. When the ambient temperate rises the resistance of thermistor decreases and current flow increases. With this feature; the ambient temperature can be monitored. These types of detectors can be used as fixed type or rate of rise heat detectors depending on design. Figure 2.10 shows the distribution of the resistance values of a thermistor according to temperature.

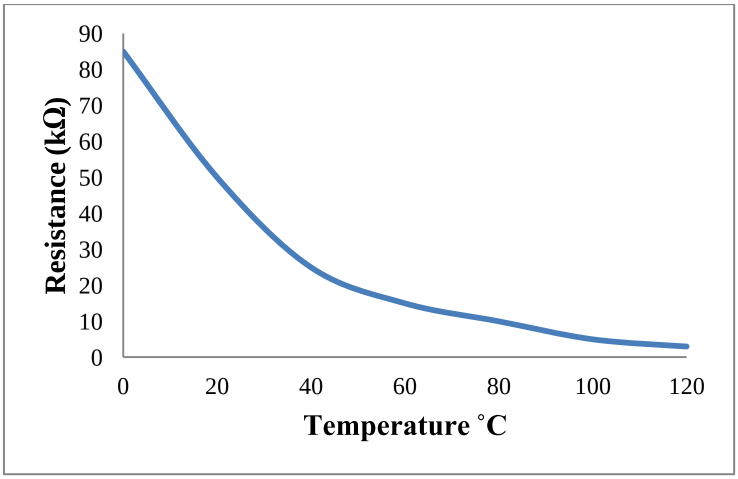


Figure 2.10. Thermistor curve

2.3. The Evolution of Robotics

The evolution of the robotics has been dominated by human necessities. First of all, industrial revolution put the robots in the factories to protect human operator from harmful tasks. The human expectation has raised in time and new robot usage areas (cleaning, construction, shipbuilding, firefighting, construction, rehabilitation etc.) and features (more flexibility, intelligence etc.) have been created to meet the human desires.

The evolution of the robotics research can be basically categorized as robot manipulators, mobile robots and biologically inspired robots (Garcia et al., 2007).

The term “Robota” is first used in Slavic languages and the original mean is monotonous work or slave labor. By the Czech playwright Karel Capek’s play R.U.R (Rossum’s Universal Robots) in 1921 the word “Robot” received another meaning. In the fiction, robots served their masters but after a while they revolted against and killed him “Rossum” and also destroyed all the life. In the fiction robots had superior abilities. R.U.R created negative sense in people mind about the robots (Wallen et al., 2008).

The Russian science fiction writer “Isaac Asimov” formulated the three fundamental laws for robots. Zeroth law was added later. The perspective was in positive manner about robots. Robots were described as mechanical creature (automaton) in human appearance without feelings (Wallen et al., 2008). The laws are given below;

0. A robot may not injure humanity, or, through inaction, allow humanity to come to harm.

1. A robot may not injure a human being, or, through inaction, allow a human being to come to harm.

2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.

3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

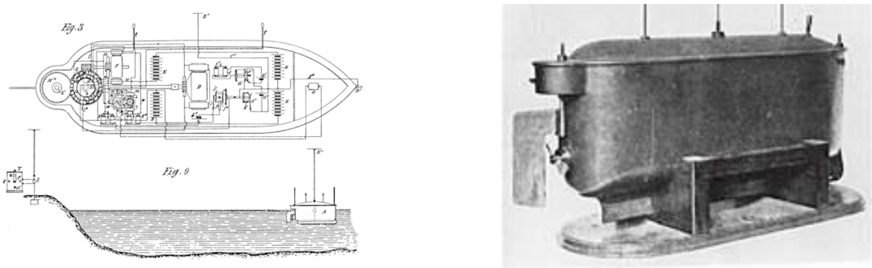
“Stig Moberg” from ABB (Asea Brown Boveri) Robotics Company completed the first law with two more laws for industrial robot. These are given below (Wallen et al., 2008);

4. A robot must follow the trajectory specified by its master, as long as it does not conflict with the first three laws.

5. A robot must follow the velocity and acceleration specified by its master, as long as nothing stands in its way and it does not conflict with the other laws.

According to the some researchers, the industrial robot history began with Heron in the first century BC. He thought about the open the temple doors automatically using the energy from altar fire with a device which convert the steam to rotational movement. Arabs also contributed robotic history. They were interested in manipulate the environment for human comfort. In the 1800s, a Swiss watch company built a number of automatons (older word for robot, objects that move automatically) like human like dolls (Anonymous, 2006).

“Nicola Tesla” had an industrial focus and was one of the greatest inventors of the industrial revolution. He thought that there was a direct analogy between machines and man in their mechanisms, senses and control. According to the Tesla, robots are not toys they are complex integrated and useful systems (Rosheim, 1994). He made pioneering works in robotics by his robot boat. It could be remotely controlled with radio waves. Tesla’s robot boat is shown in Figure 2.11.



(a) (b)

Figure 2.11. Tesla’s robot boat (a) patent drawing (b) structure

2.3.1. Robot Manipulators

A robot manipulator also known as robot arm is a serial chain of rigid limbs. Robot manipulators are designed to perform a task with end effector [14]. There are different definitions for the robot manipulators. Some of them are given below;

According to the Robot Institute of America (Sciavicco and Siciliano, 2005);

1. A robot is a re-programmable multifunctional manipulator that designed to move materials, parts, tools or specialized devices through variable programmed motions for the performance of variety of tasks.

According to the International Organization for Standardization (Anonymous, 2012 c);

2. Manipulating industrial robot is an automatically controlled, re-programmable, multi-purpose, manipulative machine with several DoF (degrees of freedom), which may be either fixed in place or mobile for use in industrial automation applications.

3. Manipulator is machine, the mechanism of which usually consists of a series of segments jointed or sliding relative to one another, for the purpose of grasping and/or moving objects usually in several DoF.

The industrial robotics researches and works are the mixture of some fields of studies. In the mechanical engineering science, the machine is studied in static and dynamic situations. Mathematic science describes the spatial motions. Sensor and interface design tasks are executed by electrical electronics engineers. Computer science provides programs to devices to carry out their tasks (Wallen et al., 2008).

Research topics of the robot manipulator can be basically categorized as (Garcia et al., 2007);

1. Kinematic calibration
2. Motion planning
3. Control law

First research area kinematic calibration is a necessary process due to the inaccuracy of the kinematic models. This process consists of four stages. The first stage is mathematical modelling. The second one is the measurement the gap between theoretical and real model by sensors and the third one is the determination of the robot’s end effector position. Last one is the implementation.

Second research area is the motion planning in which sub goals are calculated for the finishing of the robot’s task. There are two methods in the literature for motion planning; implicit and explicit. Implicit method determines the required dynamic behavior of the robot. Explicit method defines path between the robot and target.

Third area is control law that assures the execution of the plan which is required for robot’s task. The control techniques range from PD (Proportional Derivative) and PID (Proportional Integral Derivative) to adaptive control. Force control is another issue in the control law; in the action between manipulator and environment, the contact forces of the end effector is regulated.

In 1990s, new application areas comprised such as food, pharmacy industry. To meet the request of the new industrial areas new specifications and research topics have been developed. Flexibility and artificial intelligence are the most important terms in new research areas. The main aim of the flexibility is to give ability to robot for self-adaption to the product and environment. Artificial intelligence techniques are used to provide intelligence to the robot for operating in dynamic environment and uncertainty.

One of the first industrial robots, the Unimate shown in Figure 2.12 was manufactured in United States by Unimation Company in 1961. It was patented by George Devol in 1954. It was also called programmable transfer machine that designed for material handling and first installed in one of the production line of GM (General Motors) in 1962 (Karl, 2010). In 1969 Victor Scheinman designed “Standford Arm” robot shown in Figure 2.13. It was a six-axis arm for tracking arbitrary paths in three dimensional spaces. Standford Arm increased the applicability of the robots for sophisticated applications such as assembly, welding.

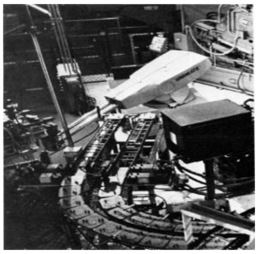


Figure 2.12. Unimate robot

Figure 2.13. Stanford arm robot

The Japanese robot industry started in 1967 when Tokyo Machinery Trading Company started to import the Versatran robot from AMF (American Machine and Foundry) Company. In 1968 Kawasaki Heavy Industries signed a license agreement with Unimation and started to produce robot in Japan. Selected milestones of industrial robotics evolution are given in Table 2.1.

Table 2.1. The selected milestones of the industrial robotics

Year	**Milestone**
1954	Devol designs a programmable factory robot (patent granted in 1961) aimed at universal automation (patent granted in 1961). His company was named Unimation.
1956	Devol's design prompts Joseph F. Engelberger to champion industrial robots and make Unimation Inc. the world's robot pioneer.
1959	A prototype Unimate arm from Unimation is installed in a GM factory. The first commercial industrial robot is installed in 1961.
1960	AMF Corp. introduces thefirst industrial robot with a cylindrical coordinate frame, the Versatran by Harry Johnson and Veljko Milenkovic.
1967	Japan imports thefirst industrial robot, a Versatran from AMF.
1968	Unimation licenses its technology to Kawasaki Heavy Industries Ltd. of Japan. This helps to ignite an explosion of robot development in Japan.
1970	Victor Scheinman at the SRI (Stanford Research Institute) introduces the Stanford Arm, an improvement on the Unimate.
1971	Cincinnati Milacron Inc. markets T3 (The Tomorrow Tool), a computer-controlled robot designed by Richard Hohn.

1973	The Asea Group of Sweden introduces the all-electric IRb 6 and IRb 60 robots designed for automatic grinding operations.
1977	ABB introduces microcomputer-controlled robots.
1978	Unimation and GM develop the PUMA (Programmable Universal Machine for Assembly) based on Victor Scheinman's robot arm design.
1979	Yamanashi University designs the SCARA (Selective Compliance Arm for Robotic Assembly). IBM and Sankyo Robotics jointly market this robot
1979	The semiconductor industry publishes the first standard for 200 mm wafers
1980	Japan becomes the world's largest robot manufacturer. By 1990, Japan's approximately 40 robot makers dominate the global robot market.
1981	Asada and Kanade build the first direct-drive arm at Carnegie Mellon University.
1984	The industrial robot industry consolidation begins. Most small robot companies go out of business within six years.
1994	The semiconductor industry plans to manufacture devices on 300 mm wafers. The first pilot line is targeted for 1997 and early production is planned for 1998 using a high level of automation.
1995	The second robot boom begins, enabled by the computer power now available. Robot–human interaction is addressed.
1997	Substrate-handling robotic systems begin operation at the first 200 mm wafer fabrication facility, by SGS-Thomson in Catania, Italy.
1997	First publication of standards for 300 mm wafer handling.
1999	SEMICONDUCTOR 300 a joint venture between Infineon Technologies and Motorola manufactures the first 64M DRAM on 300 mm silicon wafer, in Dresden, Germany.
2000	TSMC opens its first 300 mm wafer manufacturing line at a chip foundry in Taiwan.
2012	The semiconductor industry's initial target date to begin manufacturing on 450 mm wafers is 2012.

Some statistical information about industrial robots; operational stocks, usage areas and operational stocks on country basis are given in Figures 2.14, 2.15 and 2.16 respectively (Anonymous, 2012 a).

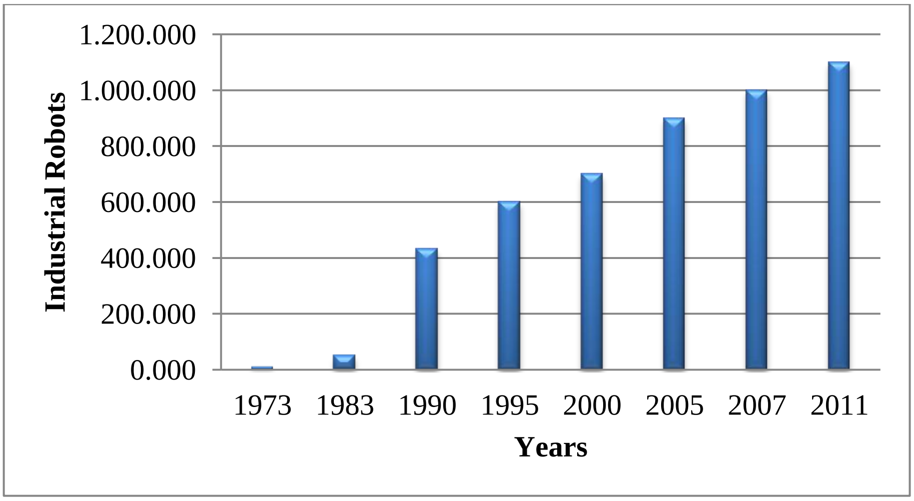


Figure 2.14. Operational stocks of industrial robots

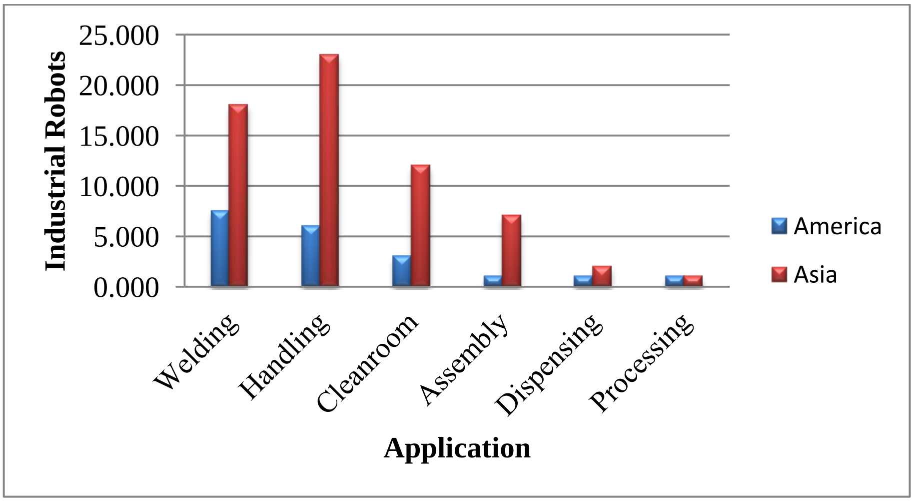


Figure 2.15. Usage areas of industrial robots

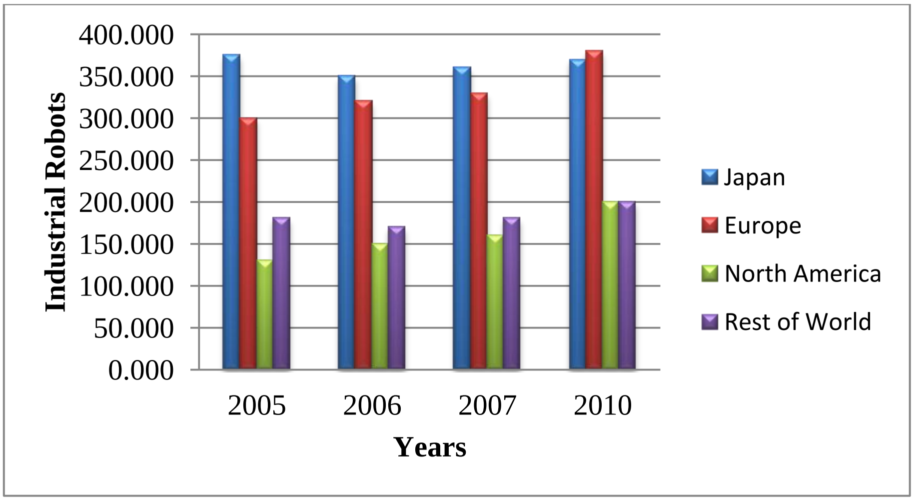


Figure 2.16. Operational stocks of industrial robots on country basis

2.3.2. Mobile Robots

The mobile robot is a system that able to complete tasks in different places. As opposed to fix based industrial robot, a mobile robot has its movement unlimited by its size due to its mobility. Mobile robots can be used to perform a variety of tasks which are normally carried out by humans such as surveillance, exploration, patrol, fire searching-fighting, homeland security, care taker, entertainer (Chen et al., 2009).

A mobile robot system consists of a platform moved by locomotive elements. The locomotive system depends firstly on the environment in which the robot will operate. These environments can be aerial, aquatic or terrestrial. Samples of the mobile robots performed in different environments are shown in Figure 2.17.


(a)


(b)


(c)

Figure 2.17. Mobile robot samples (a) aerial rescue robot (b) underwater robot (c) urban robot

In the aquatic and aerial environments, the locomotive systems are generally propellers or screws. The locomotive system in terrestrial environment is complicated. Wheels, tracks and legs are the typical terrestrial locomotive elements.

A mobile robotic system has a set of functional parts similar to human beings. These functional parts include intellectual, actuation, mobility, sensory, communication and energy (Chen et al., 2009). A model of these parts is shown in Figure 2.18.

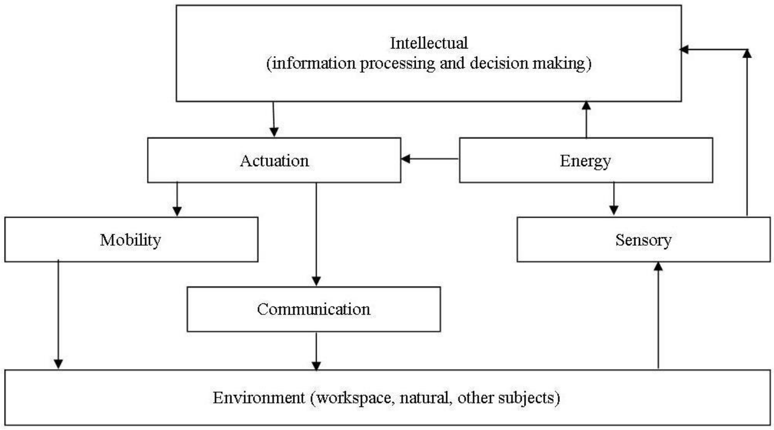


Figure 2.18. Model of mobile robot's functional parts

1. Intellectual and actuation part; controls the actuation part which drives the mobility system with information and decision process.

2. Energy; a mobile robot exerts energy onto environment. In this process, the primarily energy source (generally electrical) is converted to other types of energy such as kinetic, mechanical, wave etc. while a mobile robot carries out its task.

3. Mobility; this part consists of statue and motivational parts. Statue refers the body frame and mechanical frame of the robot. Motivational part is employed to enlarge or refine the robot mobility to execute the specific tasks. When a mobile robot carries out its task, statue and motivational parts interact with the environment. This operating environment can be classified in three categories;

    a) Pre-defined and structured environment, the robot knows the all details of (path, objects deal with) the working environment. It typically exists in factory automation.

    b) Semi structured environment, the robot has some knowledge about the environment (with GPS etc.) but the situation can be changed spatially and temporally. Surveillance robot is a typical example for semi structured environment robot.

c) Unstructured environment, the robot (for example underwater robot) has no information about the operating environment. It has to rely on its sensory and navigation system.

4. Sensory; while the mobile robot performs its task, it communicates with the environment through data, image or video. To receive the situational information, robot uses sensory function.

5. Communication; in many mobile robot applications communication part is essential. It can be employed to monitor the robot, to communicate with other robots or communicate with environment.

A functional block comparison between human beings and mobile robot (Chen et al., 2009) is given in Table 2.2.

Table 2.2. Functional block comparison between mobile robot and human being

Functional Blocks	**Human Beings**	**Robots**
Intellectual	Brain	Microprocessor
Statue	Skeleton	Mechanical frame (airframe, chassis, hull)
Motivational	Limbs	Wheels, legs, tracks, propellers, grippers, etc.
Actuation	Muscles	Hydraulic, electric, pneumatic, piezoelectric, electrostatic actuators
Sensory (perception)	Eyes, ears, skin	Cameras, optic sensors, sonar, sound, infra-red light, magnetic fields, radiation, etc.
Communication	Speech, gesture	Data, image/video, sound
Energy	Food	Power source / energy storage

The design process of the mobile robots can be categorized into three different parts;

1. Software design

2. Hardware design

3. Mechanical design

The software design is divided into two parts. Mobile robot use high level software to carry out its mission and function autonomously. Low level software contains the basic motor functions such as steering and collision avoidance.

The hardware design is twofold. Electronic parts (digital-analog and power electronic components) are employed to convert the software requests into actuator control signal and to scale, digitize the sensor signals. Actuators provide mobility to robot by converting the signal to the motion. Sensors measure the physical quantities from the environment and convert them into signal which is used for locomotion and monitoring.

The mechanical part is divided into mechanisms and body design. Mechanisms are used to transform the actuator's motion. For example; the rotational movement of the motors can be changed to the translational one. The body is designed to protect the robot from environment and it gives totality to mobile system.

The earlier mobile robots were mainly AGVs (automated guided vehicles) that generally used to transport the tools in predefined path. Today's mobile robot researches deal with autonomous applications. Research topics consist of four parts (Garcia et al., 2007);

1. Perception of the environment

2. Self-localization

3. Motion planning

4. Motion generation

The term perception can be replaced by the more general term cognition. In robotics; cognition refers the creation of the high level information from the combination of low level information. High level information can be used as a mental map of environment and prediction of the future environment. Patranik (2007) introduced a model of cognition that includes seven mental states; sensing and acquisition, reasoning, attention, recognition, learning, planning, action and coordination. In this model, the cognition is realized in three cycles; acquisition cycle, perception cycle and learning-coordination cycle. The purpose of the acquisition cycle is memorizing the information from sensors into short-term memory. Then in perception cycle the data is compared with what is known in long-term memory for validation. Finally the learning and coordination cycle plans the future actions of the mobile robot.

In structured environments, perception process allows mobile robot to generate models and maps which will be used for motion planning and generation. However, in the unstructured or dynamic environment the robot has to learn how to navigate. Therefore, the main research in mobile robotics is focusing on localization and mapping (Garcia et al., 2007).

Self-localization process allows a mobile robot to know where it is at any moment relative to its environment. For this purpose, various types of sensors are employed for measurements related with robot’s state and its environment. The localization can be local or global. In the global localization process, the mobile robot is relative to its location and orientation respect to the Global Coordinate System and the initial location isn’t necessary. In the local localization process, robot is relative to its location and orientation with respect to initial location.

Robotic mapping is the action that performed by any autonomous robot to create a map of its environment and localize itself in the map. Mapping consists of three processes (Valgren, 2007);

1. Learning the map

2. Localizing itself in the map

3. Path-planning

As localization will be easy if the robot has a map and mapping will be easy if the robot knows its localization, robotic mapping is commonly referred as SLAM (simultaneous localization and mapping) (Hahnel et al., 2003).

Although there are several markets for the mobile robots, service and homeland security-military robots are the most common types. Service robots can be categorized as professional and domestic robots. The first one is designed to serve humans or equipment. For example, medical robots can be used as assistant or trainer for surgeries. The second type domestic robot includes educational, home care, entertainment applications. The usage of remote controlled military robots is started in Iraq War to investigate the roadside bombs (Chen et al., 2009).

Historical evolution of the mobile robot (ground vehicles) started with the Grey Walter and his tortoises. He developed his first two robots; Machina Speculatrix shown in Figure 2.19 between 1948 and 1949. They were three-wheeled robots that had ability to find their way for charging (Arkin, 1998).

One of the first mobile robots with the capable of the reason, Shakey shown in Figure 2.20 was constructed in 1968 in SRI. It was equipped with a TV camera, range finder and bump sensors. Shakey was programed for perception, world-modelling and acting (Arkin, 1998).

The Stanford Cart shown in Figure 2.21 originally designed in 1970 as a line follower but it was rebuilt by Hans Morevac and equipped with 3-D vision system in 1979. In experiment; The Stanford Cart crossed a chair filled room autonomously using a TV camera which was taking picture from several angles. These pictures were used for obstacle avoidance (Nehmzow, 2003).

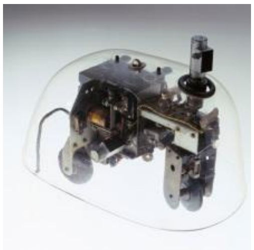


Figure 2.19. Grey Walter's turtle

Figure 2.20. Shakey robot

Figure 2.21. Stanford Cart

Genghis Khan mobile robot shown in Figure 2.22 was developed in 1988 by a robotic research group in MIT (Massachusetts Institute of Technology). This was six legged robot that designed with 12 motors, 12 force sensors and 6 pyro electric sensors. Genghis was able to learn how to scramble over boards and other obstacles (Brooks and Flynn, 1989).

The Khepera robot shown in Figure 2.23 was developed by Nicoud in 1994 in Switzerland. It was a miniature robot that had circular shape with 55 mm diameter and 30 mm height. 8 infra-red proximity sensors were used in the system as proximity and obstacle avoidance sensor (Lund and Miglino, 1996).

Stanley shown in Figure 2.24 was an autonomous car that won the 2005 Darpa Grand Challenge. This vehicle equipped with laser range finder, GPS system, 6 DoF inertial measurement unit and wheel speed measurement unit for pose estimation (Thrun et al., 2006).

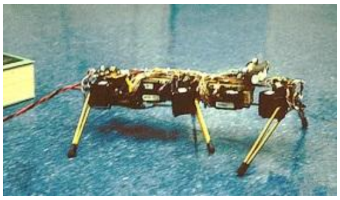


Figure 2.22. Genghis Khan robot

Figure 2.23. Khepera robot

Figure 2.24. Stanley

Some statistical information about mobile service robots; sales for professional usage between 2011 and 2012, sales forecast for professional usage between 2013 and 2016 and sales for domestic usage are given in Figures 2.25, 2.26 and 2.27 respectively (Anonymous, 2012 b).

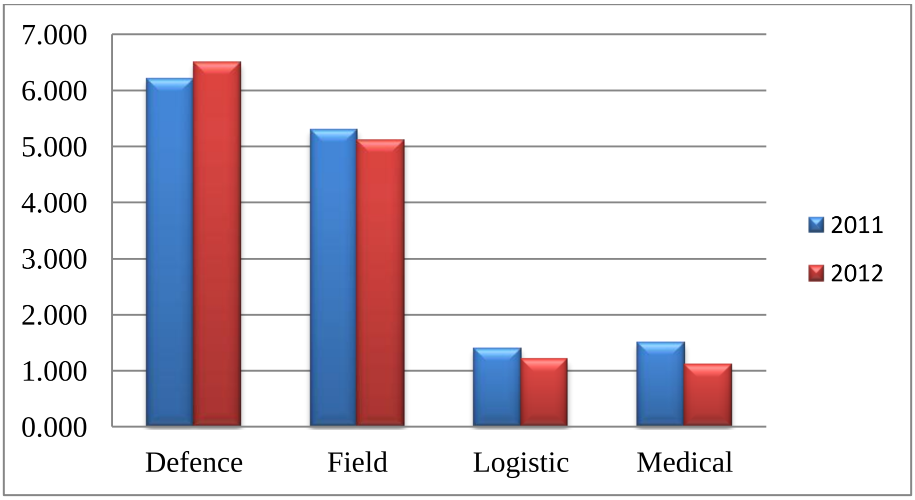


Figure 2.25. Mobile service robots sales for professional usage for 2011-2012

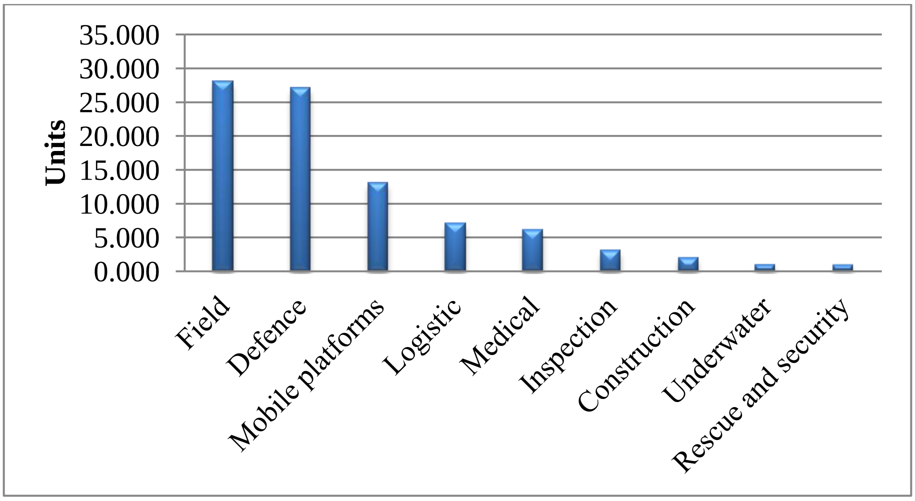


Figure 2.26. Mobile service robots sales forecast for professional usage for 2013-2016

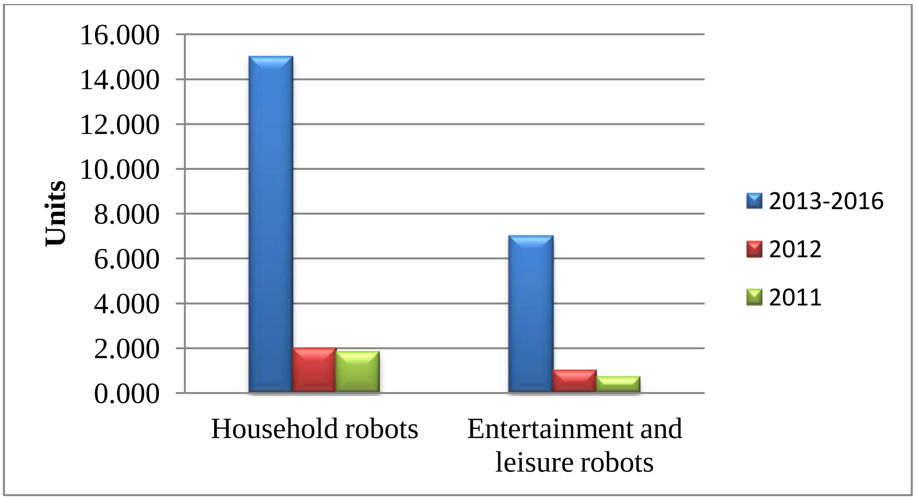


Figure 2.27. Mobile service robots sales for domestic usage

2.3.3. Biologically Inspired Robots

Apart from traditional mobile vehicles which equipped with wheels or tracks for locomotion, there is another research area, inspired from biology to produce robots with adaptive locomotion system. The most common locomotion system for biologically inspired robot is leg. The locomotion over a hard or soft surface by means of limbs or legs can be defined as walking (Ceccarelli and Carbone, 2005).

The legs of walking robots are based on two or three DoF manipulators. Walking robots have some advantages when compared with the other types of mobile robots. These are (Garcia et al., 2007);

1. Legged robots can cope with irregular terrain better than wheeled mobile robots while positioning its main body away from danger.

2. Walking robots have advantages for obstacle avoidance, stair mobility and over ditches.

3. Legged robots can walk over loose and sandy terrain easily.

4. Walking robots inflict less environmental damage than robots move on wheels or tracks.

Although walking robots have advantages, legs create a number of challenges in their own (stability and walking gait design).

Legged robot researches are focused on leg motion and coordination during robot navigation. Basically, a walking robot will be stable if it is able to keep its balance. The idea of static stability was inspired by insects and assumed the absence of inertia in the motion of the robot limbs. However, during the motion of the robot limbs some inertial effects and other dynamic components such as friction and elasticity were found to arise restricting the robot movement. Thus, researchers started to think about dynamic stability. The first dynamic stable system was developed by M. Raibert (1986) in the MIT. Until him, most of the researchers focused on statically stable multi-legged systems in order to develop dynamically stable robots (Santos et al., 2006). Raibert applied his researches into machines with two, three, four or any number of legs to solve one leg problem. In the end he succeeded. Sketch drawing of Raibert's dynamically stable system is shown in Figure 2.28.

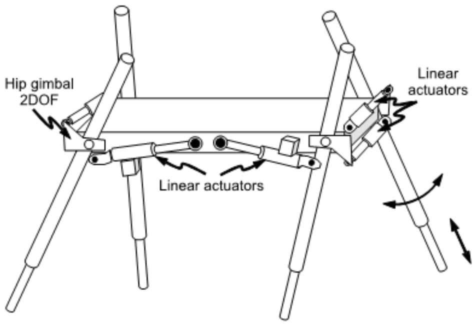


Figure 2.28. The first dynamically stable system

Another research topic for the walking robot is walking gait that directly related to the stability. Leg is a locomotion element that not continuously contact with the ground so, it is important to determine the types of gait and the sequences of leg movements (Garcia et al., 2007). There are two types of gaits depending on the stability criterion;

1. Statically stable gait (Figure 2.29-a)

2. Dynamically stable gait (Figure 2.29-b)

(a) (b)

Figure 2.29. Gait types (a) statically stable gait (b) dynamically stable gait

Statically stable gaits have the characteristic of simplifying the control of the robot with heavy limbs. They can be classified as periodic and aperiodic gaits. Periodic gaits consist of predefined sequence of movements that are repeated cyclically. Aperiodic gaits are more flexible for uneven terrain.

Legged systems have to be faster to compete with the wheeled or tracked robots. Thus, they need dynamically stable gaits. The dynamically stable gaits have been developed so far are limited with the trot, the pace and bound movements.

The interest to understand the techniques of legged locomotion observed in nature and the efforts to replicate them to mobilize have been mentioned in the mythology and ancient scripts from Greek, Indian, Egyptian and Chinese civilization (Zielinska, 2004). Mu Niu Lu Ma a wooden walking machine was built in 3rd century A.D. in China by the Zhu Ge-Liang during the preparation for the war. This machine was a walking machine that transferred its legs in a sequence when pushed, similar to a cow. It was used as a wheelbarrow for transportation of food supplies. Machine was able to cover 10 km distance in a day while carrying 200-250 kg loads. Unfortunately, no design details of this machine are available. Another significant work on automation was performed in 12th century A.D. by Badi'al-Zaman Isma'il ibn al-Razzaz al-Jazari. He compiled the text "The Science of Ingenious Mechanisms" consisted of various designs and his own inventions (Naomi et al., 2014). He also constructed an automatic machine. His machine emptied the water from basin and filled again automatically.

The first ideas about implementing the legged locomotion to vehicles were in fifteenth century. Between 1495 and 1497 Leonardo Da Vinci designed and built the first articulated anthropomorphic robot in the history of western civilization. His armored knight shown in Figure 2.30 was able to sit up, wave it arms and move its head (Silva and Machado, 2007).

Figure 2.30. Armored knight and its inner working mechanism

L.A Rygg proposed the first quadruped machine shown in Figure 2.31 named as The Mechanical Horse in 1893. This design was patented. In the design stirrups were used as pedals. The rider could power the mechanism. The movement from pedals was transmitted to legs through gears. However, there is no evidence to prove that this machine actually constructed (Naomi et al., 2014).

The Steam Man shown in Figure 2.32 was a biped machine that was proposed by Georges Moore in 1893. It was powered by 0.5 hp boiler and reached the speed 14 km/h. A swing arm was used for stability and heel spurs were employed for traction. A pressure gauge was mounted to its neck (Silva and Machado, 2007).

Bechtolsheim Baron designed and patented a quadruped machine in 1913 but there is no evidence that this machine actually was built (Silva and Machado, 2007).

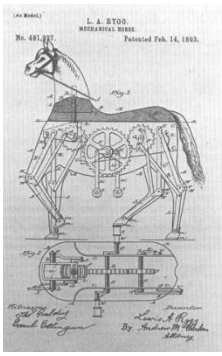


Figure 2.31. Mechanical horse

Figure 2.32. Steam man

After Second World War, walking robot research gathered a momentum due to new inventions in mechanisms, material science, electronics, control systems and computers. Many researchers started to study and develop new modern systematic walking machines in 1950s (Garcia et al., 2007).

Ralph Mosher built a four legged walking truck shown in Figure 2.33 also known as General Electric quadruped with a project started in mid 1960s and finished in

1968. Each leg of this machine had 3 DoF; one is in knee and two is in the hip. Each DoF was actuated through a crank by hydraulic cylinder. Walking machine had dimensions; 3.3 m height, 3 m long and 1,400 kg weight. It was powered with a 68 Kw internal combustion engine. General Electric quadruped controlled with 4 joysticks by well-trained operators. Despite the hardness of the control, this invention was so important for the modern walking robots. It was able to surpass the obstacles in difficult terrains (Kar, 2003).

In 1966 McGhee and Frank developed the first fully computer controlled walking robot Phony Pony shown in Figure 2.34 (Reeve and Hallamr, 2005). It had two DoF in each leg. Joint coordination was performed by the modern digital computer. Joints were actuated by electric motors through a worm gear speed reduction system. Phony Pony was powered externally through a cable. It was able to walk with two different gaits; the first one was quadruped walk and second was trot (Naomi et al., 2014).

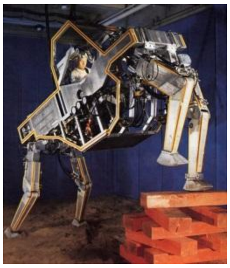


Figure 2.33. General Electric quadruped

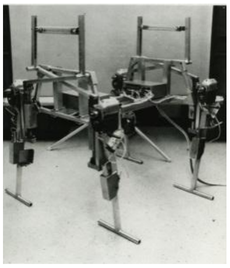


Figure 2.34. Phony Pony

In 1969, Bucyrus-Erie Company developed Big Muskie shown in Figure 2.35. It was designed for use in open-air coal mine. It is the biggest off-road walking machine has been built so far and weighing about 15,000 tons. Big Muskie had four hydraulically actuated legs that raised the body and moved forward-backward one stride and then lowered the body on the ground. When the body remained on the ground, legs lifted and moved the machine to next position. This motion was cycled by electronic sequencer (Silva and Machado, 2007).

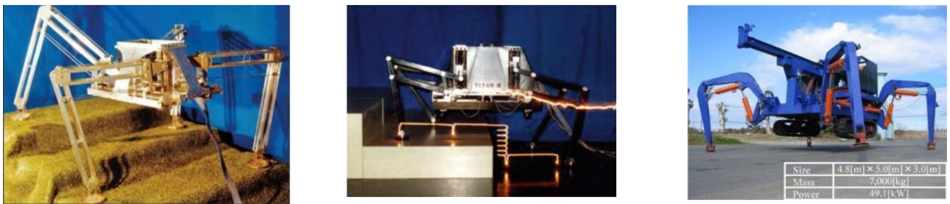
The OSU Hexapod shown in Figure 2.36 was built in 1977 as an experimental vehicle for locomotion, man-machine interaction and terrain adaptability studies in Ohio State University by McGhee and his working team. Each leg had three joints which were driven by individual motors. Every joint was equipped with one potentiometer and one tachometer to feedback joint angle and joint rate information. OSU Hexapod was also equipped with force sensors, gyroscope, proximity sensors and a camera system. A PDP-11/70 computer was used to provide the computational power to robot for real time control (Ozguner and Tsai, 1985).

Figure 2.35. The Big Muskie

Figure 2.36. OSU Hexapod

Hirose and Umetani developed a four legged walking robot PV-II shown in Figure 2.37-(a) in Tokyo Institute of Technology in 1980. It weight was only 10 kg and could walk at very low speed only 0.02 meter per second. PV-II was powered only 10 W power supply due to its weight, sophisticated leg design and low speed average. The leg actuation was carried out by DC motors and power screw speed reduction system. Its control system enabled it to maintain a horizontal body orientation.

PV-II was equipped with contact sensors on each foot to detect obstacles on the path (Naomi et al., 2014). PV-II succeeded for the first time in the world climbing the stairs based on the sensor based motion control system using the tactile and attitude sensors. Further, the TITAN-III shown in Figure 2.37-(b) was developed as the larger version of the PV-II in 1984 by Aruku Norimono in Tokyo Institute of Technology. It was equipped with intelligent program to generate adaptive walking in terrain. Its leg length was 1.2 m. Weight was 80 kg and it was built using composite material (Hirose and Kato, 2000). The latest Titan series robot was TITAN-XI shown in Figure 2.37-(c) built in 2008. It was 7,000 kg hydraulic quadruped robot that developed for drilling holes to reinforce steep slopes with rock.



(a) (b) (c)

Figure 2.37. Tokyo Institute of Technology robot series (a) PV-II (b) TITAN-III (c) TITAN-XI

The main challenges for the development of autonomous walking robot can be summarized as (Naomi et al., 2014);

1. Non-availability of energy efficient actuators; although there are some actuators with high performance (high torque and speed), weight to torque

ratio and volume to torque ratio are still problems to build energy efficient robots.

2. Reliable and economical sensors,

3. Lightweight but mechanically strong materials for construction and mechanism,

4. Computers with fast and high computing power,

5. Lightweight and onboard power source for long duration energy.

Actual autonomous humanoid (biped) robot first appeared in 1967 by Vukobratovic et al. dermato-skeletons experiment (Garcia et al., 2007). In 1973 the first controller based full scale anthropomorphic robot WABOT-1 shown in Figure 2.38-(a) was constructed in Waseda University by Kato et al. It was able to communicate with a human in Japanese and measure the distance and direction of the objects using external receptors. WABOT-1 could grip and carry the objects using the hands which were equipped with tactile sensors (Lim and Takanishi, 2006).

Kato et al. also developed WABOT-2 shown in Figure 2.38-(b). It could play an electronic organ and read the music. It was equipped with a hierarchical system of 80 microprocessors and wire-driven arms. Its legs had 50 DoF (Behnke, 2008).

(a) (b)

Figure 2.38. Waseda robot series (a) WABOT-1 (b) WABOT-2

In 1986 Honda began ASIMO Robot Project. After ten years in 1996 Honda introduced first product P2 robot shown in Figure 2.39-(a). It was the first self-contained full body humanoid robot. P2 was able to not only walk on the flat floor but also climb the stairs. P2 robot was followed by P3 shown in Figure 2.39-(b) in 1997. At the end of 2002 ASIMO robot shown in Figure 2.39-(c) was developed. It had 34 DoF, 54 kg weights and 130 cm total height. ASIMO was able to walk and run in straight or circular paths, go up or down the stairs and perform the vision routines such as object tracking, identification, recognition, localization and obstacle avoidance (Behnke, 2008; Duran and Thill, 2012).

(a) (b) (c)

Figure 2.39. Honda robot series (a) P2 (b) P3 (c) Asimo

Another important milestone of the humanoid robot evolution Sony Dream Robot was introduced in 2000 called as Qrio by Sony Company. It was a fully dynamic humanoid robot that was able to carry out human like tasks such as dancing and walking. It also recognized the faces, expressed emotion through speech and body language (Gebbert, 2014).

The humanoid robot research topics can be summarized as below (Garcia et al., 2007; Behnke, 2008);

1. Bipedal locomotion
2. Emotion expression and perception
3. Safe human-robot interaction
4. Dexterous manipulation
5. Learning and adaptive behavior

There are two main approaches for bipedal locomotion. The first one is based on the ZMP (Zero Moment Point) theory introduced by Vukobratovic. According to this theory, ZMP is a point on the ground that the sum of the moments of all active forces equal to zero. If the ZMP is within the support polygon of all contact points between the feet and ground, the humanoid robot will be dynamically stable (Vukobratovic and Borovac, 2004). Honda Asimo and Sony Qrio are the prominent robots that rely on ZMP based control (Behnke, 2008). The 2006 version of the Asimo with ZMP based control was able to run with 6 km/h speed. However; its gait didn’t look human being and Asimo couldn’t store the energy in elastic elements. Furthermore, it could only climb the certain stairs.

The second approach is to utilize the robot dynamics. The possibility of the walking down a slope without actuators and control was showed by McGeer in 1990. McGeer studied two elementary passive walking models from a wagon wheel. Rimless is on a slope and synthetic model is on level ground. These models are shown in Figure 2.40 (Narukawa et al., 2010).

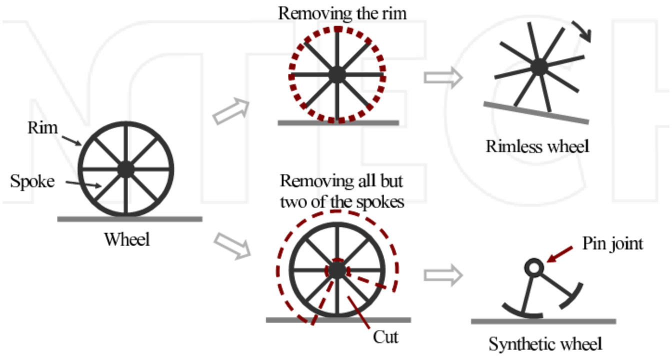


Figure 2.40. Two passive walking models of McGeer

In the first model McGeer showed that the rimless wheel has a periodic motion for a given slope angle whose stable region is very large. If the initial rolling speed and slope angle are sufficient, the rimless wheel will never falls forward. This feature is used to strengthen the stability of passive walkers. At the synthetic model, a pin joint and a large point mass were put at the hub. If the leg mass is assumed to be negligible when it compared with hub mass, the swing leg motion will not disturb the stance leg motion. The stance leg rolls at a constant speed on the floor because it is the part of the wheel. Following these analysis, McGeer increased the complexity of the biped model and developed several physical walkers with or without knees (Hobbelen and Wisse, 2007). One of them is shown in Figure 2.41.

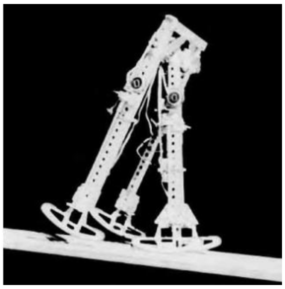


Figure 2.41. McGeer's straight passive dynamic walker

These studies have been used for useful actuated bipedal and passive dynamic walking and then actuated machines have been built.

Humanoid robot has to perceive its own and environmental states to act successfully. For proprioception, robots measure the state of their joints using encoders, force sensors or potentiometers. Robot attitude estimation is important for balance. This job is done through accelerometers and gyroscopes (Behnke, 2008). Many humanoid robots have ability to measure the ground forces at the hands and fingers and some of them are equipped with force sensitive skin. CB^2 is one of them. CB^2 shown in Figure 2.42 is a child-like robot that developed by Takashi et al. in Osaka University. It is a humanoid robot and has similar size with a child boy. It is about 130 cm high and 33 kg weights (Minato et al., 2007). The whole body of this robot is covered with soft silicon skin and many tactile sensors.

Although some humanoid robots are equipped with superior human senses such as laser range finder, ultrasonic distance sensors the most important methods for the perception are vision and audition (Behnke, 2008). Generally, movable or fixed stereo cameras and onboard computers are used for visual perception. However, a real-world image sequence is not a solved problem. These systems work well only in simplified environment. On board microphones are used for sound perception. The major problem is the separation of the interested sound source from other sound sources and noise. Because of described difficulties in perception, tele operation systems have been developed. In these systems, signals are captured by the robot and interpreted by human. Geminoid robot shown in Figure 2.43 developed by Ishiguro et al. is an example for tele operation systems. This humanoid appears and behaves highly similar to a person. It is not only controlled by an autonomous program but also manually. By introducing the manual control, the limitations in perception can be avoided and long term intelligent human-robot interactions are enabled (Nishio et al., 2007).

Figure 2.42. CB^2 robot

Figure 2.43. Geminoid robot

The general idea of the human-machine interaction is evolved from our culture human-human interaction. This interaction contains modalities such as speech, eye gaze, gestures, body language etc. In order to carry out these modalities humanoid robots with interaction abilities are equipped with expressive animated heads (Behnke, 2008). Kismet sociable robot shown in Figure 2.44-(a) developed in MIT is the one of the example. Kismet is the world's first robot that truly sociable. People can make eye contact with Kismet, read its mood from speech, and communicate with gestures (Richardson, 2008).

Robots with anthropomorphic arms and hands can be used to generate gestures. The generated gestures of humanoid robots contain symbolic gestures such as greeting, waving and pointing. The robot head can be also used for pointing and shaking (Behnke, 2008). Hubo robot shown in Figure 2.44-(b) developed in KAIST (Korea Institute of Science and Technology) equipped with articulated fingers is the one of the examples that robots can use sign language. Its height 125 cm and weight is 55kg. It can imitate the human motions such as walking, hand shaking and bowing. Hubo can move its fingers and eyeballs independently. It has 2 DoF for each eye (camera pan and tilt), 7 DoF for each hand and 1DoF for each finger (Park et al., 2005).

The most extreme form of communication humanoid robot android shown in Figure 2.44-(c) is a humanoid android that developed in Osaka. It looks like a Japanese woman and its skin is covered with a kind of silicone. Repliee Q2 has humanlike feel and neutral temperature. Forty two sensitive tactile sensors are mounted under the android's skin. It is driven by air actuators which supply 42 DoFs. Repliee Q2 can generate a wide range of motions and gestures (Matsui et al., 2005).

(a) (b) (c)

Figure 2.44. Humanoid robots (a) Kismet (b) Hubo (c) Repliee Q2

Dexterous manipulation is the one of the important capability of human beings. A human hand has about thirty DoF that it is too hard to reproduce its flexibility and sensitivity. Dexterous manipulation not only requires capable hands but also hand-arm coordination and coordination of two hands (Behnke, 2008). Shadow hand developed by Shadow Robotic Company is the one of the advanced among the robotic hands. It's driven by 40 air muscles and equipped with position and force sensors, sensitive tactile sensors on fingertips (Anonymous, 2013 d).

Humanoid robot has to have ability to learn new skills quickly to adapt daily life. In contrast to statistical learning approaches, new methods help robots to learn new skills and tasks quickly with interactive training techniques and demonstrations such as imitation-based learning and tutelage (Breazeal et al., 2006).

The time evolution of the robotics research from 1960s to 2005 (Garcia et al., 2007) is shown in Figure 2.45.

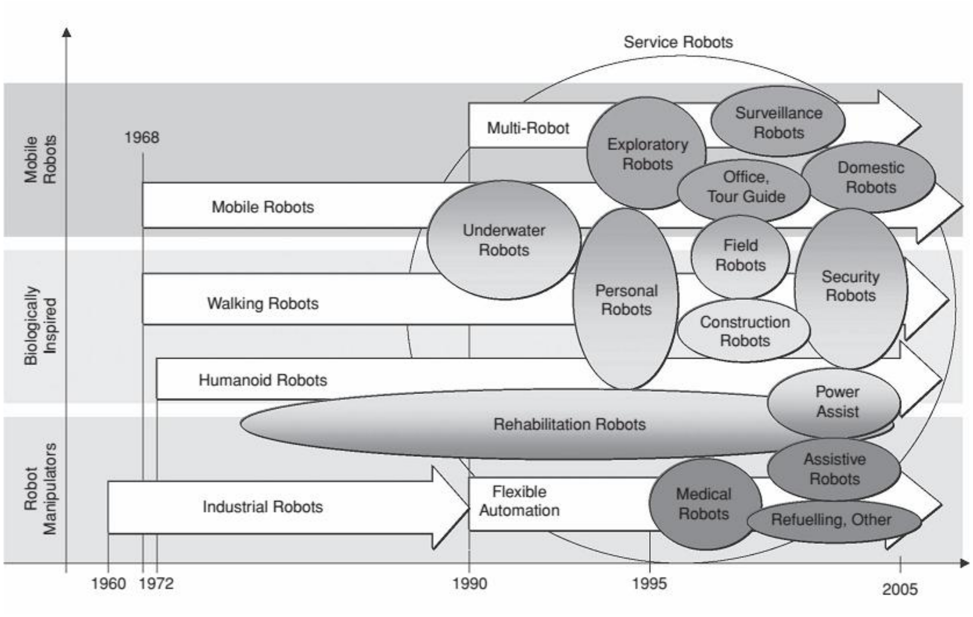


Figure 2.45. Time evolution of the robotics

2.4. Related Works (Firefighting Robots)

LUF 60 (Anonymous, 2011 a) is manufactured in Austria that powered with a diesel engine. This machine has air blower and water beam fog to clear hazardous obstacles and extinguish the fire. LUF60 is designed to use in rail tunnels, aircraft hangers, parking garages, chemical plants. The monitor nozzle is able to reach a flow rate up to 3,000 liters per minute and distance to 80 meters. It has rubber track system to strengthen mobility in high temperature condition that can operate to 230 ˚C. The robot can climb the stair, operates on slopes maximum 20 degrees. LUF60 firefighting robot is shown in Figure 2.46.

Figure 2.46. LUF60 firefighting robot

FIREROB (Tan et al., 2013) is a firefighting robot that can be controlled with remote controller and used to search in fire scene. This machine has heat shield to protect body from high temperature. It is equipped with high pressure water mist extinguisher to control the fires. It has thermal camera and sensors to observe and to monitor the fire place. FIREROB is shown in Figure 2.47.

Figure 2.47. FIREROB firefighting service robot

FFR-1 firefighting robot (Tan et al., 2013) is manufactured in United States to help the firefighters to carry out their missions. It is controlled with a remote controller. Robot is designed to use under hazardous conditions such as high temperature, poisonous materials and inside unsafe buildings. FFR-1 is suitable to operate in confined spaces, narrow streets, industrial buildings, stores, tunnels, airports, military installations, power plants and chemical plants. It has an internal double walled cooling system against overheating caused by high external temperature. This machine is able to climb up to 30 degree incline. FFR-1 is shown in Figure 2.48.

Figure 2.48. FFR-1 firefighting robot

Firemote 4800 (Anonymous, 2013 a) is manufactured in United Kingdom. This machine is an electrically powered UGV that equipped with 4,800 liter/min capacity firefighting nozzle, high pressure water system, color navigation cameras, thermal imaging camera, four gas monitors, local cooling system and roll flat hose inlet. This vehicle has an insulated stainless steel body which carries all of the equipment and reflects the radiant heat to protect the body from high temperature. Robot has also circulating water and fixed nozzle system to protect sensitive parts and cool down the body. Firemote is controlled with a panel by the help of camera system which provides environmental information. It is designed to especially operate in potential collapse areas such as workshops, factories, power plants, refineries, aircrafts, tunnels and roads. Firemote 4800 is shown in Figure 2.49.

Figure 2.49. Firemote firefighting robot

MVF-5 (Anonymous, 2013 c) is a multifunctional firefighting robot that manufactured by Croatian company DOK-ING to control the fires in unreachable areas and life threatening conditions. It is a remote controlled machine that operated with GPS-INS (Global Position and Inertial Navigation) System. MVF-5 extinguishes the fires with high-pressure cannon on hydraulic arm which pumps the water up to 55 meters away without intervention of firefighters. This machine has a high temperature resistant shield and fireproof coating to protect the system from external high temperature conditions and big flames. MVF-5 is able to withstand 700 ˚C for 15 minutes or 400 ˚C for 30 minutes. It has capability to carry 2,200 liters water and 500 liters foam tanks. Usage areas of MVF- 5 are oil refineries and terminals, military storages, chemical plants and nuclear power plants. MVF-5 is shown in Figure 2.50.

Figure 2.50. MVF-5 firefighting robot

JMX-LT50 (Tan et al., 2013) is a Chinese Robot that equipped with long distance remote control system, spray water system and barrier ability. This machine is made up of remote controller, chassis, liquid-control fire monitor and control system. It is able to protect itself from high temperature by automatic mist spraying system. JMX-LT50 uses wheel tire moving structure for traction to overcome different kind of obstacles in various conditions. JMX-LT50 is shown in Figure 2.51.

Figure 2.51. JMX-LT50 firefighting robot

SACI 2.0 (Anonymous, 2012 d) is first introduced in Brazil in 2006. This machine can be controlled wirelessly. It has 8,400 liters per minute water-foam mix pumping capability as solid or fog up to 60 meters. It has also two foam generators with 25 liters capability. Robot can pump fluids for three hours without recharging and operate full load capacity up to six hours. The other features of this machine; own lighting system, modular construction, mounted battery charging system, turbo system for increasing the pulling power at critical points, obstacle overcoming system and mobile water cannon. SACI 2.0 is shown in Figure 2.52.

Figure 2.52. SACI 2.0 firefighting robot

ArchiBot-M (Anonymous, 2013 f) is Korean firefighting robot that designed to use in human inaccessible location. The main purpose of machine is checking and clearing the paths for fire fighters. The robot is equipped with special suspension system for stair climbing and working in high temperature. It has also waterproof ability and cooling system. ArchiBot-M is shown in Figure 2.53.

Figure 2.53. ArchiBot-M firefighting robot

Thermite T2 (Anonymous, 2013 b) is manufactured in United States to be operated in hazardous material fires, forest fires, chemical plant fires, rail fires. Robot is controlled remotely up to the 400 m distance. Thermite is designed to operate in rough terrain. The other features of this machine are; high temperature durability, A440f steel and 5th grade aluminum construction, 5 seconds startup time for robotic functions and 2,200 liters per minute water pumping capacity. Thermite T2 is shown in Figure 2.54.

Figure 2.54. Thermite T2 firefighting robot

MyBOT 2000 (Tan et al., 2013) is Malaysian remotely controlled firefighting robot. It is made up of a mobile and rigid chassis. This machine is wirelessly controlled. Nozzle system of this robot can be directed at different angle and can be elevated in order to control the fire at different height. Robot uses electric energy for power. It is equipped with state of art sensor and imaging system to detect and locate fire victim. MyBOT 2000 is shown in Figure 2.55.

Figure 2.55. MyBOT 2000 firefighting robot

Luo and Su (2007) developed an intelligent security system for buildings which contains autonomous navigation, master slave operated system, supervision through internet, a remotely operated camera vision system, danger detection and diagnosis system. This system is able to receive the building environmental status (fire, smoke, intruder and gas) and send information to user using internet and GSM (Global System for Mobile). The intelligent security system contains an ISR (intelligent security robot), remote supervisory computer, security module and appliance control module. The robot has cylindrical shape with 50 cm diameter and 140 cm height dimensions, produced using aluminum frame. ISR made up of upper and lower bodies. Upper side carries an IPC (industrial computer) with the features; Pentium-III 933CPU (Central Processor Unit) and 256 MB RAM (Random Access Memory), touch screen, charge coupled device (CCD) camera, sensors and sensory circuit. Lower body carries the drive system, batteries and two DC servo motors. ISR is shown in Figure 2.56.

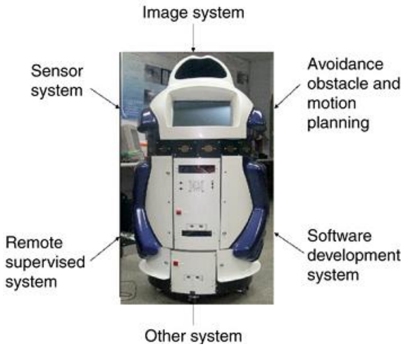


Figure 2.56. Intelligent security robot

The ISR can receive environmental status (dangers such as fire, gas etc.) from security module and transmit to appliance module through wireless RF interface. Robot can interact with GSM module using RS232 interface and communicate with remote supervisory computer through internet.

The multiple sensors and fuse sensory data are used in system to detect the fire and generate reliable fire detection signal. Ionization smoke sensor, UV flame sensor (R2868) and temperature sensor (AD590) are employed for fire detection.

Fire detection tests of the ISR are shown in Figure 2.57.

(a) (b)

Figure 2.57. Fire detection tests of the ISR (a) flame detection status (b) gas leak detection status

Khoon et al. (2012) developed an AFFMP (autonomous firefighting mobile platform) that patrols and monitors the prescribed area for searching the fire occurrence with flame sensors and extinguish it. Robot consists of microcontroller system, fire detection unit, navigation and line tracking module and fire extinguishment unit. The AFFMP robot is shown in Figure 2.58.

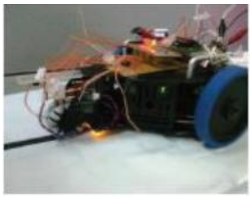


Figure 2.58. Khoon's autonomous firefighting robot

The flame detection is built around flame sensors which have capabilities to detect flames from 760nm to 1100nm wavelength. The front flame sensing system is designed to keep AFFMP 10 cm distance away from fire source. ADC (analogue to digital converter) receives the output of flame sensors and then, transmits them to input ADC pin of microcontroller. When flame sensor value is greater than the predefined threshold value, system decides that there is fire source in front of AFFMP.
A dynamic method is used to set minimum threshold value for different environments;

1. The AFFMP captures the light intensity of any environment for 10 times without any fire source,

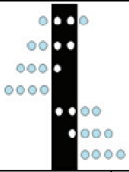
2. The maximum and minimum data are eliminated (Median Average Filtering Method),

3. The remaining 8 values are used to calculate the average value,

4. An offset value is added to average (delta T=4) so that the random interference is taken into account.

Four pairs of LED (light emitting diodes) and LDR (light dependent resistor) are used for line tracking. These sensors are installed on the front part of AFFMP body chassis symmetrically namely as left sensor, middle left sensor, middle right sensor and right sensor. By default LDR sensors will yield logic '1' since white background reflects back all the emitted light, sensors outputs will be logic '0' when black line is detected. The principle of line tracking is given in Table 2.3.

Table 2.3. Line tracking methodology



Case	Left sensor	Middle left sensor	Middle right sensor	Right sensor	Necessary movements
1	1	0	0	1	Forward
2	1	1	0	0	1 unit right
3	1	1	1	0	2 units right
4	1	1	1	1	3 units right
5	0	0	1	1	1 unit left
6	0	1	1	1	2 units left
7	1	1	1	1	3 units left

A DC fan is used for fire extinguishment which can blow off the candle flame. An ultrasonic sensor is also employed for avoiding obstacles. Experimental test results of the Khoon's robot are given in Table 2.4.

Table 2.4. Test results of Khoon's robot

Types of Sensors	**Detection Range**
Flame Sensor	Rated specification : 20cm (4.8V)-100cm (1V) Through experiments: In front of candle source at about 15cm, 3.6V to 4.1V.
Outputs of LDR Sensors	At a height of about 1.2cm between LDR sensors and the surface, when the line track is not detected, the voltage levels are; Right : 3.41V Middle Right : 3.39 V Middle Left : 3.37V Left : 3.36V When black line track is detected, the voltage level will be ZERO.

Chang et al. (2006) developed a FSR (fire searching robot) using task oriented design (TOD) methodology. This system is designed to operate especially for indoor fires in the areas less than 165 m^2. TOD is a schematic robot design method that based on the results of the analysis of the tasks. The main goals of the FSR are;

1. To become aware of location of the victims, fire points and to send these information to the firefighters,

2. Stable mobility in a hazardous and uncertain environment,

3. Operator safety and convenience in operation.

The mobile element of FSR has three DoF, two DoF are used in right-left track operation and third one is for auxiliary track arm. FSR can climb the stairs using the left-right tracks. Chang's fire searching robot is shown in Figure 2.59.

Figure 2.59. Chang's fire searching robot

Robot has three special features. These are;

1. The system is operated by wireless remote control so that the safety of the operator is guaranteed.

2. The operator can watch the image acquired by the attached IR camera. With this function firefighter can easily determine the victim and fire position.

3. With the mobile elements robot can climb the stairs and work in narrow spaces.

The hardware of the robot controller system consist of Intel 1.4 GHz embedded pc (personal computer), DAQ (data acquisition board) (Sensoray S626), a bluetooth module and 900 MHz RF (radio frequency) module. The technical specifications of FSR are given in Table 2.5.

Table 2.5. Technical specification of Chang’s fire searching robot

Items	**Specification**
Total Size	900 x 650 x 370 mm
Weight	100 kg
Payload	70 kg
Maximum velocity	0.5 km/h
Gradability	30°
Operating time	1 hour
Traction motor type	DC Motor
Maximum range of wireless remote control	200m

Kim et al. (2009) developed a portable fire evacuation guide robot system that can monitor indoor disasters. It can be used for victim detection and atmosphere observation. System contains voice communication module and LED guidance lamps for victim rescuing operation. Several robots can communicate in this system for searching in large areas with WPAN technology (wireless personal area network). System is designed to operate in proposed fire scenario which shown in Figure 2.60.

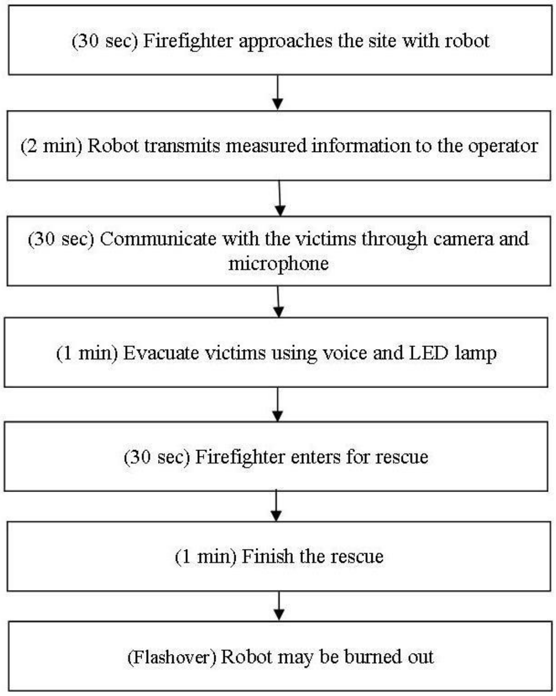
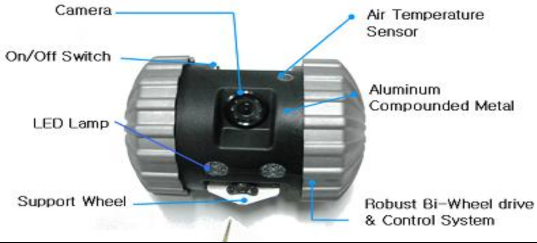


Figure 2.60. Kim's fire and operating scenario

After firefighters carry the robot system in the time taken about 30 seconds, they deploy their robot by throwing into the fire site. The system starts to gather internal information such as temperature, CO, O_2 gases. All gathered information about site situation, video and voice data is transmitted to firefighter operator using wireless channel. When firefighters detect the victims, they can communicate with them using microphone and speaker system.

The robot has spool-like round shape with 120 mm x 120 mm x 120 mm dimensions and 2 kg weights which makes it easy to carry. The main body of the evacuation robot consists of two DC motors, a RC (radio controlled) servomotor, a camera, sensors (temperature, CO, O_2, gas, compass), a speaker, a microphone, LED-set lamp and RF module. It has also an embedded operating system. Design and hardware specifications of the portable fire evacuation guide robot are given in Table 2.6.

Table 2.6. Hardware specification of the Kim’s robot



Category	**Specification**
Voltage range	6–9 V
Electric current (operation mode)	200 mA
Electric current (standby mode)	50 mA
Battery type	LiPo
Maximum movement speed	282 mm/sec
Weight	2 kg

The external body of the robot is manufactured with aluminum compound metal. Internal circuit is produced using teflon wiring that enables the system to survive up to almost 250 ˚C. The electronic circuits of the robot are inserted into the body and whole outer case is covered with waterproof epoxy adhesive. For the impact resistance; leaf springs are inserted into the gears and robot’s wheel. Outer impact from the vertical direction is decreased by leaf springs with internal coil. The side bolster also distributes the diagonal external impact over the whole body. System can distribute and resist the external impact. Impact resistance mechanisms are shown in Figure 2.61.

(a) (b)

Figure 2.61. Impact resistance mechanisms of Kim’s robot (a) leaf spring (b) side bolster

The impact resistance and waterproof features of the system are tested in 9m x 3m x 2.3m room. The impact resistance is tested throwing the robot to the earth with increasing falling heights until external cracks are observed. The maximum undamaged height is 3.5m for throwing. Waterproof (cooler) feature of the robot is tested operating the system under falling water situation. High temperature protection of the robot is tested both in electronic oven and real fire condition. It is observed that; the robot is suitable for monitoring the sites and rescuing victim in early stages of fire before the ambient temperature reaches flashover.

Xu et al. (2011) developed a mobile robot remote fire alarm system. The system consists of the mobile robot and remote terminal. While mobile robot acquiring and sending the information about fire scenario, remote terminal receives this information and alarm. The main functions of the mobile robot and remote terminal are given below;

Mobile robot functions:

1. It not only detects the fire on both sides of the fixed route, but also detects the fire in the area without fixed route.

2. It can detect the smoke.

3. It can sense the environment temperature.

4. Robot stops and alarms when there is an obstacle closer than 50 cm in front of it. (Obstacle avoidance).

5. Robot is able to record the distance traveled.

6. It can transmit the information wirelessly in real time.

Remote terminal functions:

1. It can receive the data sent from mobile robot in real time and alarm.

2. It displays the fire information and save them in real time.

Mobile robot is made up of STC89C52 MCU (microcontroller unit) as control center, far IR flame sensor, E18-D80NK IR obstacle avoidance sensor, smoke sensor, DS18b20 temperature sensor, photoelectric sensor and SRWF-1021 wireless transmission module. Remote terminal uses STC89C52 microcontroller as control center, LCD12864 (liquid crystal display), clock module and wireless transmission module.

The drive mode of the mobile robot is differential drive. Four units of LED and IR reception diodes are used for line (fixed route) tracking. Three units of far IR sensors are used for flame detection. They are arranged as left, right and center sensors. Robot uses the right and left units to drive forward to fire, center sensor to determine the distance from fire. When the distance reaches about 10 cm the robot stops, sends information to the remote terminal and alarms. The fire detection range can be up to 2.3 meters.

Dual power supply is used in the system. First one is A-a rechargeable nickel cadmium 14.4 V/1500mAh battery that supplies LM7805 (voltage regulator) which offers 5V output voltage for sensors and controller. Second one is B-a rechargeable nickel-cadmium 7.2V/ 2000mAh battery that supplies power for motors.

Zhang et al. (2012) developed an intelligent inspection robot using combined flame sensors and IR sensors. The system combines the flame and IR sensors to avoid obstacles while searching for high temperature flames. Hardware of the robots is divided into three modules; sensing and communication, motion and power.

Sensing and communication module consists of PSD (position sensitive detector), LPC2368 model process chip, two CAN (controller area network) communication interface, an extra AD port, three expanded I/O (input-output) interface and one expanded 5V power supply. In this module; the analogue signals from IR sensors are received and transformed into digital signal so that the distance between the object and robot is calculated. PSD sensors have ability to measure the distance in the range of 10-80 cm with the ± 30% accuracy error. Arduino sensors are used in the system as flame detector. Flame detection is carried out in the range of 760nm–1100 nm wavelength, working temperature is between -25 °C and 85 °C and probe angle is 60°. Lithium batteries are used as power supplier. L298 N motor driver and two DC motors are used to establish the motion module.

Roberto et al. (2013) proposed a multisensor data fusion technique for fire detection through a mobile robot. Detection system is based on temperature, luminosity measurements and flame detection.

The robot has 19x11 cm dimensions and operates based on tread mill which is suitable for inhospitable environment. Tamiya DC engine, Ardumoto driver and 3.7V battery are used in the motion system. Robot is controlled by Arduino 2009 board. System is shown in Figure 2.62.

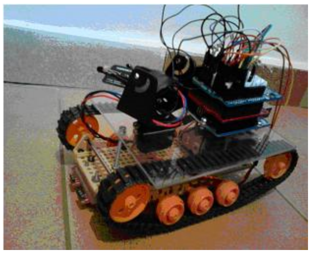


Figure 2.62. Roberto’s fire detection robot

The data fusion algorithm of the system is operated based on moving average filter method. The moving average point is calculated using the average values of the temperature, luminosity and flame sensors collected in a time interval. Fire detection tests are executed for alcohol and paper flames. Temperature and luminosity detection, flame sensor behavior in different distances from fire and the best moving average parameters are tested.

Moving average parameters are tested at 30 cm distance from the flame for 60 seconds time interval. 10, 20, 30 seconds time periods of the moving average values are evaluated. Test result for 10 seconds time period is shown in Figure 2.63.

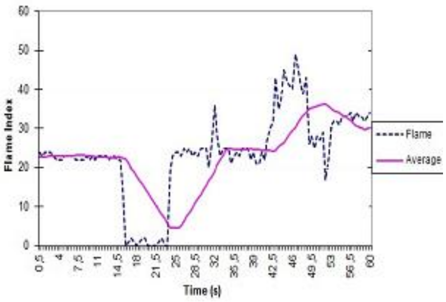
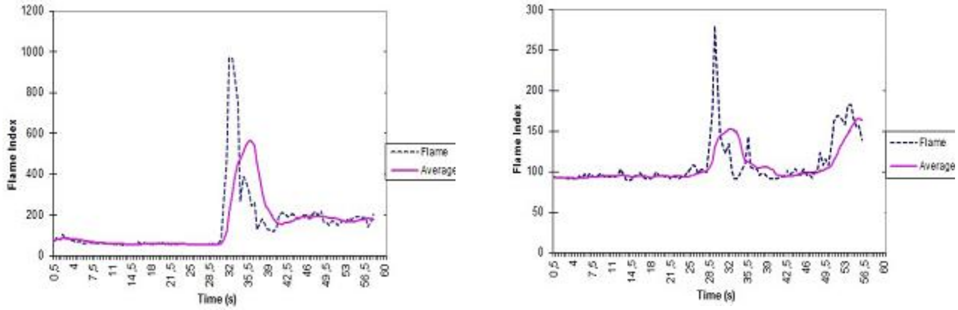


Figure 2.63. Moving average test result for 10 seconds time period

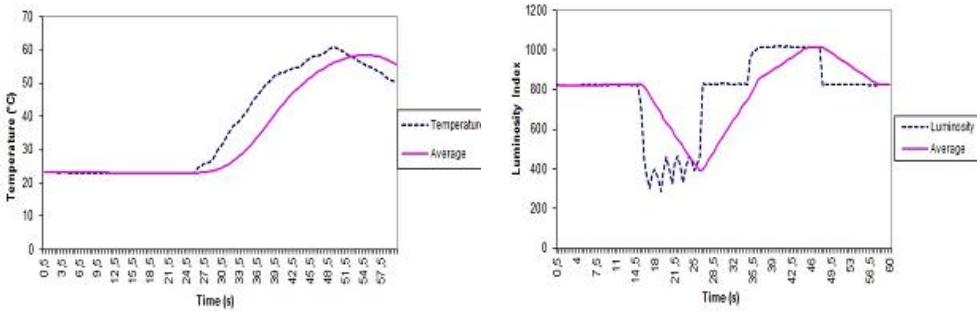
To determine the ideal flame sensor position, the distance tests between the robot and flame are carried out. As the result; it is determined that the maximum sensing distance is 1 m and minimum distance is 10 cm. 1 m distance is determined as the best choice for flame detection because of the safety reasons. Distance test results are shown in Figure 2.64.



(a) (b)

Figure 2.64. Roberto's system fire detection distance test results (a) 10 cm from the flame (b) 1 m from the flame

Threshold of the temperature and luminosity are tested. A normal luminosity value of a room without fire is supposed 700 and 830 lux. Normal room temperature is between 20 ˚C and 35 ˚C. Therefore, threshold values are indicated as 900 lux for luminosity and 40 ˚C for temperature sensors. Temperature and luminosity tests results are shown in Figure 2.65.



(a) (b)

Figure 2.65. Roberto's system (a) temperature sensor test result (b) luminosity sensor test result

Kumar et al. (2007) developed a gesture controlled robot with the capabilities of fire extinguishment, audio and video capturing. Robot is made up of fire extinguisher set and spy camera.

The system consists of Atmega 8 microcontroller, flex sensors, RF module, DC motors, relays, TV tuner card, video camera and water sprayer. Flex sensors are used to understand the finger movement. When flex sensor is bent with finger behavior the resistance value of the sensor is changed. Therefore, the motion of the robot can be controlled. TV tuner card is used by computer to receive information from camera.

Tongying et al. (2010) developed a flame detection system based on a mobile robot platform. In the system, environmental images are obtained with a camera which placed on mobile robot. An image edge detection method is proposed to suppress the noise interference and detect the flame image accurately. The system includes an exploration-rescue mobile robot which is suitable for especially coal mines. Robot consists of vision system installed on the arm, flameproof electrical engine, power supply and sensors. The hardware system of the robot contains; robot actuator, robot controller computer, sensory system, A/D collection card and human machine interface. The exploration-rescue robot is shown in Figure 2.66.

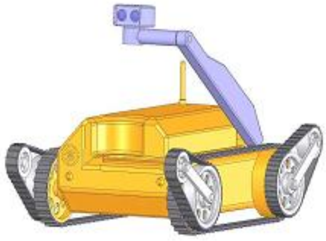


Figure 2.66. Guo’s exploration-rescue robot

Martinson et al. (2012) developed a robot (Octavia) which can be used as a team member for firefighting tasks. In the system, when human team leader indicates the location of the fire using speech-gesture and clears the obstacles, robot finds the exact location of the fire with its sensors and extinguishes the fire with CAF (compressed air foam) system.

Octavia has 48 DoF in body, arms and head. It is mounted on a two wheeled Segway base for mobility. Robot uses the perceptual system to understand what team leader means and behaves correctly. Perceptual system contains; two color cameras in eyes, Mega-Imaging SR-4000 flight camera in head, IR camera centered on the chest. A Hokuyo laser range finder is employed for the navigation. Octavia Robot is shown in Figure 2.67.

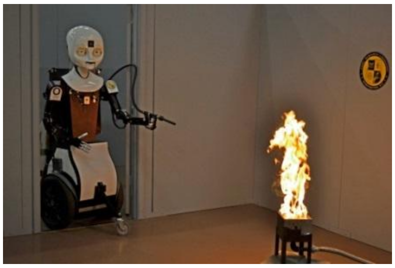


Figure 2.67. Octavia robot

SR-4000 camera and clustering approach are used to detect and track people in the environment. In the person tracking system; when the intensity image from extracted region contains a face, this region is added to the list of active tracks. The initial detection range of the robot is limited about 2 m for tracking. However, once a person has been detected, they could be tracked in 5m distance. A combination of speech, face, clothing and complexion recognition is used for person identification. Once a person has been identified, specific gesture and speech models are employed to improve recognition in difficult environments.

Speech recognition system consists of two functions; the first one is to understand speech in noisy environment. COTS (Commercial off-the-shelf) speech recognition system with a dictionary relevant to firefighting task is used to understand the speech. The second is the spatial understanding. To execute this task, the usage area is divided into smaller areas. The combination of periodic gesture and deictic gesture is used for gesture recognition. While periodic gesture is describing the terms such as follow, stop and begin for searching the fire, deictic gesture tells the approximate location of the fire. In fire localization system, robot looks for a set of places at which to aim a stream of water instead of searching for precise location of fire.

Zhang (2009) developed a remote controlled firefighting robot which operates based on a small multi-functional crawler hydraulic excavator. The capabilities of the robot are; walking, turning, striding, dangerous material transportation and fire extinguishment.

Robot can climb over any barrier up to height of 200mm. Temperature sensor is used to prevent the system from fire. If the environment temperature is higher than the limited value, the fire extinguishment system will be activated to cool the robot. A pressure sensor is installed to control the manipulator better. When an object grasped by the manipulator, sensor checks the pressure value, if this value reaches limited, system will stop to apply pressure.

According to the functions, robot consists of four systems; remote control, hydraulic system, traction mechanism and working equipment. The remote control system is composed of a transmitter and a receiver. The first control signal is generated by control handle and coded, modulated and amplified by transmitter and sent to receiver. Receiver amplifies, demodulates and decodes the signal and then sends it to electric and hydraulic control valves. The hydraulic system contains; pump, hydraulic cylinders, motor, pipelines and control valves. This system is employed to control working equipment and traction mechanism. Traction mechanism and working equipment are composed of pedrail wheel, climbing equipment and manipulator.

3. MATERIAL AND METHOD

As expressed in introduction part of thesis, it is planned to design and implement a fire detection mobile robot that is able to patrol and monitor environment, inspect and detect the fire occurrence.

The main purpose of the robot is early fire detection at industrial areas. Although, firefighting is an extremely dangerous task it still being carried out by human operator. Time is also an extremely important parameter for firefighting tasks. When fire reaches the disaster situation it is too hard to control it and human beings often cannot do anything. Therefore, early fire detection is the best solution for firefighting.

There is some fire detection robots built for early fire detection. However, they have many disadvantages. They can patrol at prescribed paths but the physical real lines are required for their operation. They often use only one of the flame, smoke and heat sensors to detect the fire. One sensor is not sufficient to detect fire and cause false alarms.

We proposed a fire detection mobile robot which has ability to patrol at virtual lines and monitor the prescribed area while scanning for fire occurrence via flame, smoke and heat sensors. There are some difficulties to design a robot which is able to execute all of these tasks. So, design goals and criteria should be indicated carefully before the manufacturing process. The design goals and criteria are listed below;

1. The first design goal is to reduce the costs and time. As the robot is designed for industrial applications high cost and time requirement is not acceptable. To reduce the cost and time; standardized, available industrial components should be used.

2. The mechanic part of the robot should be robust and scalable to carry payloads and dynamics.

3. The training periods for new users should be short so that new users can learn the mobile robot in a short time interval and produce better results. For this purpose, the complexity of the components should be avoided.

4. The velocity of the robot should be high enough to patrol at industrial areas. For the first experiment; 0.5 m/s motion speed is adequate. The proper components for moving mechanism should be selected and combined.

5. The weight of the robot should be optimized. It should be not too heavy or too light. The weight affects the power consumption and balance directly. So the carrier board material and components should be selected carefully.

6. Dimensions of the robot should be specified carefully. Carrier board should be designed carefully for good dimensioning.

7. Energy requirement of the robot is an important parameter. Energy consumption is affected by dimensions, weight and motor selection directly. Thus, the components should be selected and designed as eco-friendly.

8. The programming of the robot should be basic as possible so that understanding and changing will be easy.

9. The system should not require real physical lines for patrolling. Thus, robot can operate in different areas at different paths.

10. To detect the fire occurrence correctly; a combination of smoke, flame and heat sensors should be used. Otherwise, it causes the false alarms.

3.1. Mechanical Part

3.1.1. Carrier Board Design and Manufacturing

There are several ways to produce the carrier board. Different material types, varies manufacturing and connection methods and different connector types are available. Before the manufacturing process, the carrier board design criteria should be considered carefully. First section is the board material selection. Board material should have some properties such as light weight, easy machining and high flexibility. Wooden material was selected as carrier board material. Rigid plastic foam was also used to prepare second layer of the carrier board.

The selection of the board manufacturing and motor connection method is the second section of process. There are many methods which have been used for manufacturing and connection technology. These are categorized as fasteners-adhesives, soldering, solid state and arc welding and high energy fusion. Although, there are many options for motor connection, connection with the fasteners-adhesives is the best solution for our robot project because we want to build a modular platform which is assembled and disassembled easily. Adhesives, blind fasteners, rivets, screws can be used to connect motors. The rivet is the easiest solution because it is a cheap material and easy to find it everywhere. We used flat head type rivets with 4 mm head diameter.

The third section is the preparation of the motor-carrier board connectors. We used L shape aluminum with 30 mm width, 80 mm length and 2 mm thickness to produce the motor-carrier board connectors. Shape and dimensions of the motor connector are shown in Figure 3.1.

After the selection of the proper components, we manufactured carrier board and motor connectors easily using simple mechanical processes such as cutting, drilling. Carrier board with intended dimensions (400 mm length, 300 mm width and 5 mm thickness) was prepared using cutting process and motor connectors were connected to carrier board using drilling-screwing process.

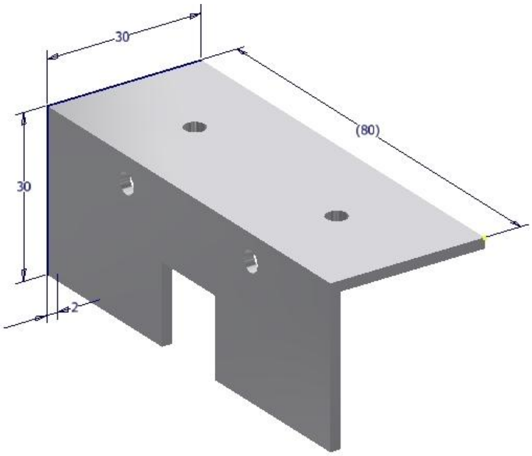


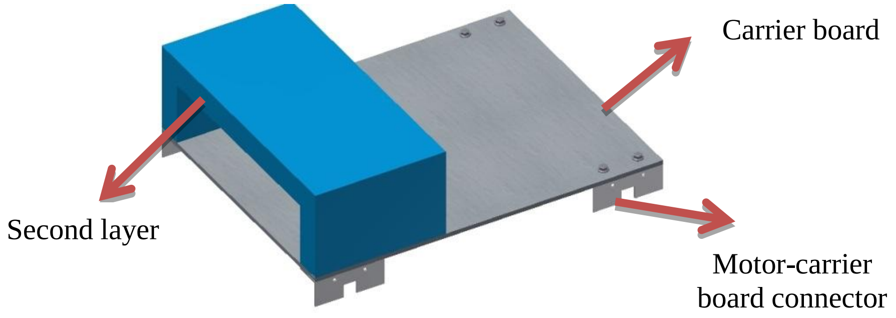
Figure 3.1. Motor-carrier board connector

3.1.2. Drawing and Dimensioning of Carrier Board

Drawing and dimensioning of the robot's carrier board is an important parameter. The dimensions of the robot affect the all other factors and parameters used in the robot design such as power requirement, placement of the components etc. The weight of the robot chassis can be calculated easily with drawing and dimensioning. Then this information can be used to calculate power requirement.

Two dimensional sketches were drawn using a CAD (computer aided design) program and solid state drawings and assemblies were prepared using a solid modeling computer program. Design of the carrier board and detailed dimensions are given in Table 3.1.

Table 3.1. The design and the dimensions of carrier board



The Dimension Type	Value of the Dimensions
Board width	300 mm
Board length	400 mm
Second layer width	300 mm
Second layer length	150 mm
Second layer height	100 mm
The total area of carrier board	0.12 m^2
The total area of second layer	0.045 m^2
Total weight of the carrier board	0.8 kg
The distance between wheels	370 mm
Ground clearance	80 mm

3.1.3. Traction Mechanism

Pallet and wheel are the most common tools used at the traction mechanism. Each one has own advantages and disadvantages. Generally; pallet provides better traction performance in rough surface but they have higher cost and harder to assemble, wheels are easier to assemble but their traction is less than pallet's. The detailed advantages of the pallet and wheel are given below (Marlene, 2014);

The advantages of the pallet

In straight pulling, the pallet system can develop more force than wheel for the same weight. Pull/ weight ratio is the ratio of pull a vehicle produces, divided by the vehicle weight. Normally the heavier vehicles produces more pull, whatever their power delivery system. Slip is the ratio of a vehicle's actual forward speed to the track or wheel speed. Normally, slip increases as the pull increases. The pallet system has advantage for pulling parameter especially in rough terrain applications.

The pallet system has advantage for optimization especially in rough terrain applications. Optimization is the selection of proper vehicle weight and traction tool set-up for a particular speed, pull and ground condition. Using a vehicle at its optimum gives the highest performance and lowest operation cost. Optimizing a vehicle with wheel for a given speed, pull and ground condition requires changes in ballast and tire inflation pressure. So, a vehicle with wheel may not be at an optimum set-up for its current working condition. With pallet traction, there are no pressures or weights to change so the traction system can perform at their optimum.

The pallet system has advantage for power delivery efficiency. Power delivery efficiency is a measure of the ability of a traction system to deliver available engine power into useful work on the ground. Pallet system can maintain their efficiency over a wide range of pull and ground conditions.

The advantages of the wheel

The vehicles with wheel generally steered better than the vehicles with pallet. When a vehicle with wheel is steered, the wheels turn and point in the direction of the turn. When a vehicle with pallet is steered; the track on the outside of the turn speeds up and the track on the inside of the turn slow down. Under heavy loads, wheel may slide sideways as it turns, but it still turns. Under the similar situation, pallet system may not turn at all. This can present a serious problem.

Steering a pallet system requires more power than steering with wheel. The extra power is needed to solve the sideways slipping and speed up the outside track. This can cause overload problems when the vehicle is steering while operating near full engine load.

As a summary;

1. Pallet provides power efficiency, better traction on the slippery surface.

2. The robots with the pallet look more aggressive than the robots with wheel.

3. Pallet systems have less ground impact.

4. The robots with pallet can support heavy load, since they spread the weight over entire surface.

5. Wheels have low production cost.

6. Wheels need lower amount of torque to power. Therefore, they can reach high speeds easier than the pallets.

7. Wheels generally lighter than the pallets.

8. Several materials can be used to produce wheels in order to meet.

9. Wheels have less moving parts. This means that there are fewer components that can get damaged.

After the consideration of the advantages and disadvantages of the wheel and pallet, the wheel type traction system was selected. Wheel type traction provides; low cost, high speed and light weight system. With these advantages we designed and built an easy to use, low cost and fast fire detection robot.

3.1.4. Wheels

The size and material of wheel is an important parameter that mobile fire detection robot can move its weight and payload efficiently. The wheel selection also affects the torque and energy requirement of the robot. We used four pieces of Pololu PL-1439 model wheels. Details of the wheel are given in Table 3.2.

Table 3.2. Details of the wheels

Diameter	90 mm
Width	10 mm
Motor connection hole diameter	6 mm

3.1.5. Motor-Wheel Connector

The motor connector is one of the mechanical parts of the robot that holds the wheel one side and motor other side. We used Pololu PL-1999 model connector for motor-wheel connection. Details of the motor-wheel connector are given in Table 3.3.

Table 3.3. Details of motor-wheel connector

	
Diameter	25 mm
Width	10 mm
Motor connection hole diameter	6 mm

3.1.6. Steering Method

There are many steering methods to control the wheels or pallets and to drive the robots. The selection of the steering method is an important parameter which affects the cost, mechanical structure, electronics and control system of the robot. Several steering methods are described below;

<u>Differential Drive</u>; is the most common steering method. The main idea is simple, velocity difference between two motors drive the robot in any required path and direction. Therefore, the name is differential drive. Differential wheeled robots have two independently driven wheels fixed on a common horizontal axis or three wheels where two independently driven wheels and a roller call attached to maintain the equilibrium. Four fundamental cases of the differential driving are given below;

1. If both wheels are driven at the same speed and same direction (either clockwise or anticlockwise) the robot will move in a straight line.

2. If the wheels rotate at equal speed but in opposite directions, both wheels will traverse a circular path around a point centered half way between two wheels. Therefore, the robot will spin in place.

3. If one of the wheels is stopped, while the other continues to rotate, the robot will spin around a point centered approximately at the mid-point of the stopped wheel.

4. If one wheel rotates faster than the other, the robot will follow a curved path and turn toward the slower wheel.

The advantage of this method; the design process, mechanical construction and control algorithm are very simple with this driving method.

The disadvantage of the differential driving method; the robot isn't driven as expected. It is neither driven along a straight line nor turned exactly at expected angles, especially using DC motors. This is due to difference in the number of rotation of each wheel in a given amount of time. To handle this problem, a correction factor should be added to the motor speed to reduce the faster wheel speed.

The fundamental cases of the differential driving method (Anonymous, 2010 b) are shown in Figure 3.2.

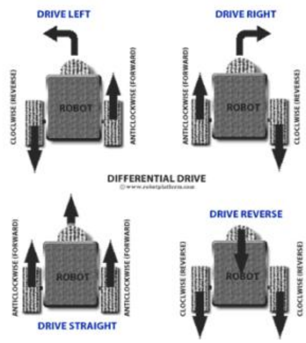


Figure 3.2. Fundamental cases of differential driving method

Skid steering; is essentially the same as differential steering except that this type of steering is executed by a tracked robot, or a robot which has multiple powered wheels in fixed (non-steerable) positions. This steering method engages one side of the tracks or wheels and turning is done by generating differential velocity. In differentially driven robot, there is castor which balances the robot but in skid steer drive, the castor is replaced with two driving wheels. Fundamental cases of the skid steering are indicated below;

1. When turning left is required; the right wheels or tracks are driven forward and the left wheels or tracks are driven backward until the robot turns left.

2. When turning right is required; the left wheels or tracks are driven forward and the right wheels or tracks are driven backward until the robot turns right.

3. When both wheels and tracks are driven at same direction, they have 360˚ turn with almost 0 angle radius and moves in a straight line.

4. When one side (right or left side) rear and front wheels or tracks rotate faster than the other side wheels or tracks, the slower side (center side) wheels almost skid to turn. Thus, the name of this steering method is skid steer.

Some advantages and disadvantages of skid steering method are given below;

Advantages:

1. This method has better traction ability than differential driving especially for rough terrain.

2. The same design method and concept can be used on both tracked and wheeled robots.

3. This method does not require castor wheel and so eliminates the problems caused by castors.

Disadvantages:

1. This method reduces the life time of the wheel or tracks since it uses skidding or slipping technique.

2. Driving in a straight path is hard to achieve as both motors will rotate at different speed when expected to rotate at exactly same speed. This problem can be handled by other sensing devices, but it adds extra cost to steering mechanism.

The fundamental cases of the skid steering method (Shamah, 1999) are shown Figure 3.3.

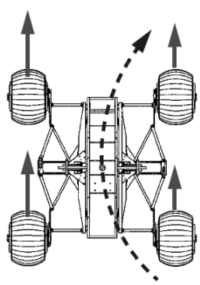


Figure 3.3. Skid steering method for multi-wheeled robot

Ackermann steering; is the one of the most common steering configuration that also used for robotic steering. Ackermann steering mechanically coordinates the angle of two front wheels which are fixed on a common axle used for steering and two rear wheels fixed on another axle for driving. The advantages of this steering method can be summarized as below;

1. The increased control

2. Better stability and maneuverability on the road

3. Less slippage and power consumption

The Ackermann steering configuration (Hrbacek et al., 2010) is shown in Figure 3.4.

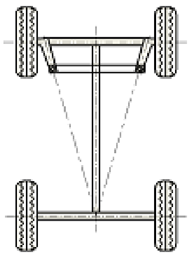


Figure 3.4. Ackermann steering method

Ackermann steering is designed to avoid the slippage problem. To handle the problem; when there is turn, the inner wheel turns with a greater angle than the outer wheel and reduce the slippage. The Ackermann approach is generally used for fast outdoor robots which require excellent ground clearance and traction.

<u>Omni directional drive;</u> Omni directional robots are built using Omni wheels and castors. As Omni wheels have smaller wheels which attached perpendicular to the circumference of another bigger wheel, they allow robots to move in any direction instantly. The major advantage of the Omni directional drive is that the wheels of the robot do not need to turn to move in any direction. The robots with Omni directional drive can move in any direction without changing their orientation. Generally Omni wheeled robots are equipped either three or four wheeled platform. Each of them has its advantages and disadvantages. Two types of the platform with three and four wheels (Anonymous, 2010 b) are shown in Figure 3.5.

(a) (b)

Figure 3.5. Omni directional platforms (a) three wheels (b) four wheels

<u>Independent drive;</u> in this steering method, each wheel is driven and controlled explicitly. The problem with this method is the coordination between the wheels as each wheel heads in its own direction. When the coordination fails, it affects the entire system. When the coordination between the wheels is achieved, this method is the best choice for uneven and untested environment, since one of the wheels stops working; the remaining wheels can pull the robot to desired position and direction. The independent type of steering method (Anonymous, 2010 b) is shown in Figure 3.6.

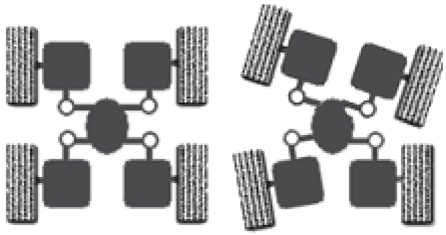


Figure 3.6. Independent drive

After the consideration of steering methods; the differential steering method was selected for fire detection robot. The use of differential steering system makes the robot flexible enough for planned application (semi-structured environment, level area). This steering method requires the lowest number of motors, mechanic and electronic components. Differential steering provides low weight, long operation time and less power consumption advantages. With these advantages; we could design and build a less complicated, low cost and fast enough fire detection robot.

3.2. Hardware

In hardware section, the details of the hardware elements of fire detection robot are described. First, the required power and torque for the robot motion is calculated. After the calculation; the used motor set and battery unit are explained.

3.2.1. Power Requirement Calculations

For the calculation of the minimum required power, the maximum estimated total force and the desired speed must be known. The maximum total force is equal to sum of resistance forces. The resistance forces are; gradient force, air force (is assumed zero), rolling and inertial force which is equal to mass times acceleration.

$$F_{Total} = F_{gradient} + F_{air} + F_{rolling} + M_m \, x \, a_{max} \quad (3.1)$$

Where:

F_{Total}	Maximum estimated total force
$F_{gradient}$	Gradient resistance force
F_{air}	Air resistance force
$F_{rolling}$	Rolling resistance force
M_m	Total mass of the fire detection robot platform
a_{max}	Maximum acceleration

The forces acting on the fire detection robot are described in Figure 3.7.

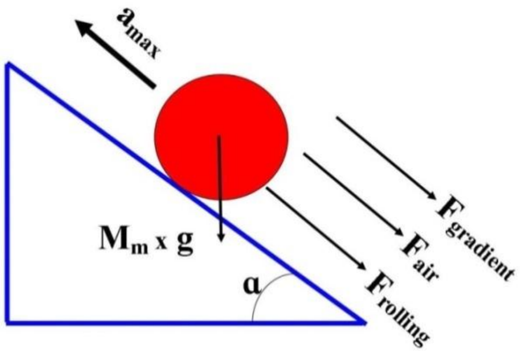


Figure 3.7. Resistance forces

Gradient resistance force is the first calculated force. Gradient force is the component of the weight of the robot that is parallel to its path. The maximum force occurs when the robot climbs the maximum appointed angle. The maximum road angle is determined as 10˚ for our application. The estimated weight of the robot is 2.5 kilograms (robot and notebook weights).

The formula of the gradient resistance force is;

$$F_{gradient} = M_m + \sin a \times g \tag{3.2}$$

Where:

a	Angle of the slope that robot climbs
g	The acceleration of the gravity

The result of the (3.2) for 2.5 kilograms estimated robot weight and 10˚ slope is;

$$F_{gradient} = 2.5 \text{ x } \sin 10^{\circ} \text{ x } 9.81 \approx 4.25\ N \tag{3.3}$$

The second calculated force is the rolling resistance force. The rolling resistance mainly changes due to the deformation of the road and surface of the wheel.

The formula of the rolling resistance force is;

$$F_{rolling} = f_r \text{ x } W \tag{3.4}$$

$$W = M_m \text{ x } cos\ a \text{ x } g \tag{3.5}$$

Where:

f_r	Coefficient of the rolling resistance
W	Wheel load

The rolling resistance surface mainly depends on the rolling resistance coefficient and the slope angle that fire detection robot operates. The rolling resistance coefficient (Naunheimer et al., 2011) is selected from the Table 3.4.

Table 3.4. Coefficent of rolling resistance

Road surface	**Rolling resistance coefficient (f_r)**
Smooth tarmac road	0.010
Smooth concrete road	0.011
Rough, good concrete surface	0.014
Good stone paving	0.020
Bad, worn road surface	0.035
Soil	0.070-0.240
Loose sand	0.150-0.300

Rolling resistance coefficient is specified as 0.020 from the table. Using the formula (3.4) and selected coefficient; the rolling resistance force is calculated as:

$$F_{rolling} = f_r \text{ x } W = f_r \text{ x } cos\ a \text{ x } g = 0.020 \text{ x } 2.5 \text{ x } \cos 10^\circ \text{ x } 9.81 \approx 0.5\ N \tag{3.6}$$

The last resistance force comes from the acceleration. It is assumed that the fire detection robot has an acceleration of 0.2 m/s^2 at the maximum power required condition. The total resistance force can be found with the results of the equations (3.3), (3.4), maximum acceleration value and the determined mass of the robot.

$$F_{Total} = F_{gradient} + F_{air} + F_{rolling} + M_m \text{ x } a_{max} = 4.25 + 0 + 0.5 + 2.5 \text{ x } 0.2 \quad (3.7)$$

$$F_{Total} = 5.25\ N \quad (3.8)$$

After the calculation of the total resistance force, the required power output of the robot can be calculated by multiplying the total resistance force with maximum operational speed of the fire detection robot. The required power is calculated as:

$$P_M = F_{total} \text{ x } V_{max} \quad (3.9)$$

Where:

P_M	Required power output of the motor
V_{max}	Maximum speed of the robot

Speed for the fire detection robot is determined in the range of 0.2 m/s and 0.5 m/s. For the power calculation, the maximum speed value is assumed as 0.5 m/s. With the equation (3.9) required power is calculated as:

$$P_M = F_{total} \text{ x } V_{max} = 5.25 \text{ x} 0.5 = 2.625\ W \quad (3.10)$$

We designed our robot with four steered wheels. So the calculated power requirement value is for four motors in our application. As a result, the power requirement of a single wheel is about 0.66 W.

3.2.2. Torque Requirement Calculation

The torque requirement is another important parameter to select the proper motors. Total weight and diameter of the wheels must be known to calculate torque requirement. The selected wheels have 90 mm diameters.

The formula of the torque requirement is;

$$T = [M_m \text{ x } g \text{ x } (\sin a + f_r) \text{ x } d\ ] / 2 \quad (3.11)$$

Where:

T	The torque of the axis
d	The wheel diameter, in meter

The result of the (3.11) for 2.5 kilograms estimated robot weight and 10˚ slope is;

$T = [M_m \times g \times (\sin a + f_r) \times d\,] / 2 =$

$[2.5 \times 9.81 \times (\sin 10^\circ + 0.020) \times 0.09] / 2 \approx 0.21\ Nm$ (3.12)

As a result, the torque requirement for a single motor is about 0.053 Nm.

To calculate the power and torque requirements of the system a “Force-Torque Calculator” was developed using C#. Screenshot of this calculator is given in Appendix-3.

3.2.3. Motor Selection

After the calculations of power and torque requirement, a brushed DC motor set was selected for robot motion. It is a proper and affordable motor when the criteria such as size, weight, price, power, torque and velocity outputs are taken into consideration. It is also easy to find this type of motor in any industrial place.

The motor rotates at 120 rev / min and powered with 12V DC. The rated power is 32 W with a torque of 2.6 Nm. Maximum power requirement of single motor was calculated about 0.66 W and torque requirement was about 0.053 Nm. The technical properties of the motor can easily satisfy the robot’s power and torque requirements. Selected motor is shown in Figure 3.8.

Figure 3.8. 12 volt 120 rpm DC motor

As mentioned before, our maximum speed expectation is 0.5 m/s. The relation between the angular velocity and linear velocity can be calculated as;

The formula of the relation is:

$V_{max} = r \times w \times 0.10472$ (3.13)

Where:

r	The radius of the wheel
w	Angular velocity of the motor, in RPM (Rounds per minute)

$V_{max} = r \times w = 0.045 \times 120 \times 0.10472 = 0.566$ m / s (3.14)

The calculated velocity of the motor can satisfy our speed expectation from the robot.

3.2.4. Energy Supply

Fire detection robot needs energy supply to execute its tasks. The battery configuration selection is an important parameter in the design and implementation process. The dimensions, voltage, weight and recharge method are the important factor to select the battery type.

There are many types of rechargeable batteries such as lithium-ion, lead-acid, lithium polymer and nickel-cadmium. LiPo type of rechargeable battery was selected as energy source. The advantages of LiPo are indicated below (Anonymous, 2012 e);

1. There are low to high rate cells are available with capacities generally as little as 20 mAh up to 4000 mAh.

2. LiPo batteries have a greater energy density in terms of weight.

3. LiPo batteries provide higher volumetric density with very thin cells (under 5 mm).

4. There is high flexibility in cell sizes and shape with LiPo batteries.

5. LiPo batteries offer superior stability in over voltage and high temperature conditions.

6. Nowadays, many high tech electronic devices such as smart phones, GPS devices and netbooks employ LiPo cells.

A LiPo type of battery unit with three cells as shown in Figure 3.9 was used in the system. The battery has 12V voltage and 2.2 Ah capacity specifications. System can be operated for one hour without recharging. As a result, battery unit can satisfy the energy requirement of the motors and electronic components.

Figure 3.9. LiPo battery unit

3.3. Obstacle Avoidance (Ultrasonic Sensors)

Obstacle avoidance is an important feature of the fire detection robot that it can detect the obstacles and calculate the distance from them. Robot can move freely without any collision with any obstacle using this calculation. Obstacle avoidance is done with different technologies such as ultrasonic sensor, IR sensor, and camera. Each of these technologies has its own advantage and disadvantages.

We used two HC-SR204 ultrasonic sensors (Anonymous, 2011 c) shown in Figure 3.10 that provide 2 cm to 400 cm non-contact measurement. Its accuracy can reach to three millimeters. The sensor module includes ultrasonic transmitter, receiver and control circuit.

Figure 3.10. Ultrasonic sensor

The working principle of the sensor can be summarized as below;

1. The module uses the at least 10 μs (microsecond) high level signal for trigger.

2. The sensor sends eight 40 kHz sound signals and detect whether there is a signal back.

3. If the signal back, the time of the output signal duration is the time from sending ultrasonic to returning.

Ultrasonic ranging module has 4 pins;

VCC	5V supply
GND	Ground
Trig	Trigger pulse input
Echo	Echo pulse output

The distance calculation formula is;

$$D = V \times t \qquad (3.15)$$

$$D = 1 / 29 \times t / 2 \qquad (3.16)$$

$D = t / 58$ (3.17)

Where:

D	The distance from obstacle
V	Speed of the sound 343.2 m/s at 21 °C air
t	The echo time

The sound speed = 3, 4320 cm /1,000,000 s = 1/29 cm per µs

The time is divided with two, since the ultrasound sensor measures the sending and returning times.

The features of the ultrasonic sensor are;

Working voltage	5V DC
Working current	15 mA
Working frequency	40 Hz
Maximum detection range	4 m
Minimum detection range	2 cm
Measuring angle	15˚
Trigger input signal	10 µs
Dimensions	45x20x15 mm

3.4. Fire Detection Unit

The most commonly used sensors for fire detection are smoke and heat sensors. We designed the fire detection unit using flame sensor, temperature sensor and smoke sensor because one sensor for fire detection cannot provide accurate fire detection information. For example, a smoke sensor can sense the fire when it is so close to fire source. Therefore, we need to use multiple sensors to detect a fire incident and fuse sensory data to generate reliable fire detection.

3.4.1. Flame Sensor

The flame sensor is an IR receiving transistor. It uses the characteristics of the IR rays which are sensitive to the flame. With a special IR receiving tube the sensor can get the flame rays and then, turns the brightness of the flame into level signals to input of the processor.

We used Arduino flame sensor (Anonymous, 2013 i) shown in Figure 3.11 which is suitable for 5V voltage working environment. It can detect light source in the range of 760 nm-1100 nm wavelength. The detection distance is to 100 cm. This sensor's outputs can be digital or analog.

Figure 3.11. Arduino flame sensor

Arduino flame sensor has 4 pins;

VCC	5V supply
GND	Ground
DO	Board digital output interface (0 and 1)
AO	Board analog output interface

The features of the flame sensor are;

Wavelength range	760 nm-1100 nm
Detection distance	20 cm (4.8V)-100 cm (1V)
Detection angle	15°
Operating voltage	3.3V-5V
Dimensions	3x1.5x0.5 cm
Weight:	8 g

3.4.2. Temperature Sensor

We used LM 35 Centigrade temperature sensor (Anonymous, 2011 d) shown in Figure 3.12 by Dallas Instruments in the system as temperature sensor. It is a precision integrated circuit temperature sensor with an output voltage linearly proportional to the Centigrade temperature. It provides ± 0.25 °C accuracy at room temperature and 0.75 °C accuracy over full range of -55 °C to 150 °C temperature.

Figure 3.12. LM 35 temperature sensor

LM 35 has 3 pins;

VS	5V supply
GND	Ground
Vout	Voltage output

The features of the flame sensor are;

Calibration	Directly in Celsius Centigrade
Scale factor	20 cm (4.8V) – 100 cm (1V)
Measurement range	-55 °C to 150 °C
Operating voltage	4V to 30V

The sensor circuit is shown in Figure 3.13.

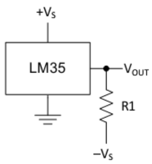


Figure 3.13. Circuit of temperature sensor

With 10 kΩ (kilo-ohm) R1 resistance value, LM 35 sensor can measure the temperature at full range. Full range measurement values are given in Table 3.5.

Table 3.5. Full range measurement values of LM35

Vout	**Temperature**
1500 mV	150 °C
250 mV	25 °C
-550 mV	-55 °C

3.4.3. Smoke Sensor

The fire smoke consists of CO (carbon monoxide), CO_2 (carbon dioxide), S (sulfur), NO (nitrogen oxide) and water vapor. Therefore, smoke detection task can be carried out by detecting one of these gases.

We used Pololu MQ-9 model CO and flammable gas sensor (Anonymous, 2013 g) shown in Figure 3.14 as a smoke sensor in the system. It is a semiconductor gas sensor that can detect the presence of carbon monoxide at concentrations from 10 to 1,000 ppm (parts per million). The sensor can operate at temperatures from -10 ˚C to 50 ˚C. Sensitive material of the MQ-9 is SnO_2 (tin oxide). It can make detection by the method of high and low temperature. Sensor detects CO when heated by 1.5V (with low temperature) and detects combustible gases such as methane, propane when heated by 5V (with high temperature).

Figure 3.14. MQ-9 carbon monoxide and flammable gas sensor

The sensor circuit of MQ-9 sensor is shown in Figure 3.15.

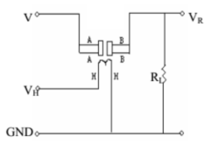


Figure 3.15. Circuit of smoke sensor

Two voltages; VH (heater voltage) and VC (test voltage) are required for sensor operation. VH is used to supply a working temperature to the sensor. VC is used as VRL (detect voltage) on RL (load resistance). VC and VH can use the same power circuit.

The MQ-9 sensor was used with Pololu Carrier board (Anonymous, 2013 h) shown in Figure 3.16. This board simplifies the interface from 6 pins to 3 pins. Three pins are; ground, power and analog voltage output. Board has two mounting holes and provides convenient pads for mounting the sensor's required-setting resistor.

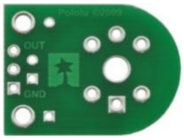


Figure 3.16. Carrier board

The technical features of the MQ-9 sensor are:

Consumption	≤ 150 mA at 5V
Heater voltage	5.0V ± 0.2V (for high), 1.5V ± 0.1V (for low)
Heater time	60 ± 1 second (for high), 90±1 second (for low)
Load resistance	Adjustable
Heater resistance	31Ω ± 3Ω at room temperature
Heater power consumption	≤ 350 mW

3.5. Fire Detection Robot Platform Configuration

As mentioned before, wooden material and rigid plastic foam were chosen to build carrier board of the robot. Motors, battery unit and data acquisition and control circuits were mounted to wooden layer of the carrier board. Second layer was used to carry the netbook. Ultrasound sensors, fire detection unit and servo motor were mounted to front side of the carrier board. The general assembly of the robot platform is shown in Figure 3.17

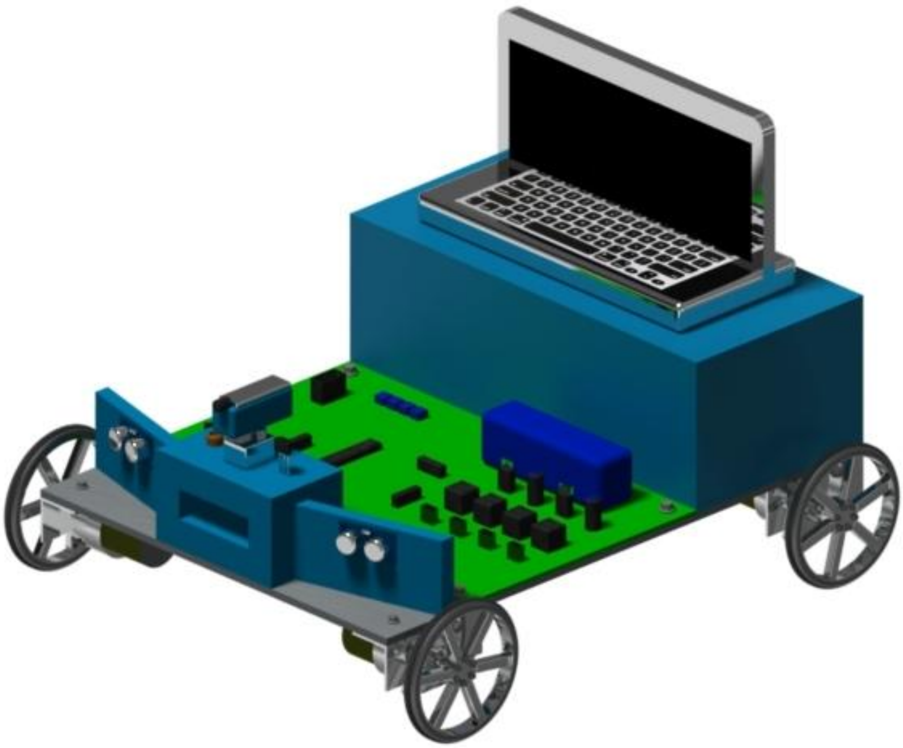


Figure 3.17. General assembly schematic of the robot

The disassembly schematic of the system which shows the detailed configuration and components of the robot platform is shown in Figure 3.18.

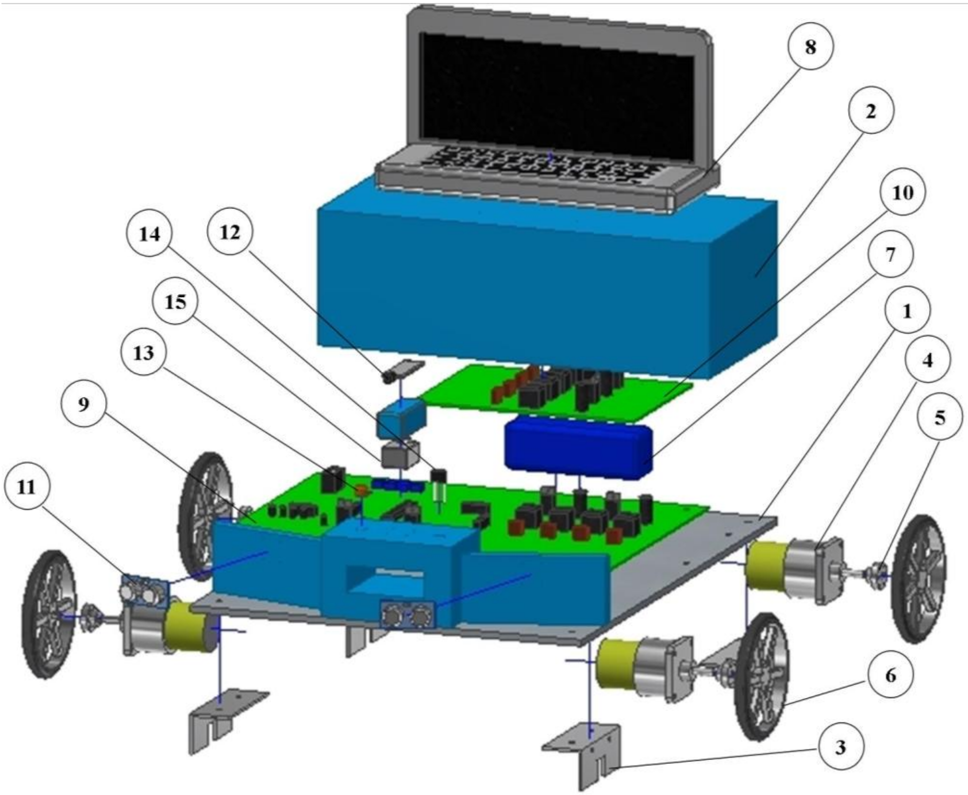


Figure 3.18. Disassembly schematic of the fire detection robot design

Names and functions of the components numbered on Figure 3.18 are shown in Table 3.6.

Table 3.6. Names and functions of the components

Number	Name	Function
1	Carrier board	Carrier board was used to mount and carry the other components such as battery unit, data acquisition and control circuits and fire detection unit.
2	Second layer	Top plate was produced using plastic rigid foam because of its manufacturing ease. Netbook was mounted to second layer.
3	Motor-carrier board connector	Motor-carrier board connector was used to connect motors to carrier board.
4	Motor	Brushed DC motor was selected. It was used for differential steering system of the robot.
5	Motor-wheel connector	Motor-wheel connector was used for motor wheel connection.
6	Wheel	Four pieces of wheels were used as steering wheel in the system.
7	Battery	LiPo type of battery unit was mounted to carrier board. It was used as energy source.
8	Netbook	Netbook was used to get data from the data acquisition unit of the robot.
9	Data acquisition and control circuits	Data acquisition and control circuits were designed and produced for special requirements of the robot instead of buying a commercial card. It was used to get data from sensors (fire detection unit, obstacle sensors) and control the servo motors. These circuits were mounted to carrier board.
10	Motion control unit	Motion control card was designed and produced to control the brushed DC motors.
11	Ultrasonic	Two pieces of ultrasound sensors were employed for obstacle avoidance. They were mounted to front side

	sensor	of the carrier board.
12,13,14	Fire detection unit	Fire detection unit consists of flame, smoke and temperature sensors. Flame sensor was mounted on servo motor which is able to rotate 180˚.
15	Servo motor	A servo motor was used in the system to provide 180˚ rotating ability to flame sensor.

3.6. Data Acquisition and Control Circuits Design and Implementation

Data acquisition and control circuits were designed and produced for special requirements of the robot, instead of buying a commercial card. These circuits were designed using Proteus Software. This software consists of two modules. The first one is Isis and second is Ares. Isis module was used to design and simulate the schematic diagram of the data acquisition and control circuits (these circuits are given in Appendix-2), after schematic diagram, Ares module was used for printed circuit design.

Data acquisition and control circuits design and implementation process was categorized into four parts;

1. DC motors control unit
2. Data acquisition and communication unit
3. Servo motor control unit
4. Microcontrollers

3.6.1. DC Motors Control Unit

Motor control unit was designed to control the speed and the direction of the rotation of the motors. The electronic components of the motor control unit are listed below;

1. Relays
2. 1n5822 diodes
3. IRF 3205 Power MOSFETs
4. Capacitors
5. ULN 2003 Darlington Transistor Array (Relay driver)

Relays; Four pieces (two is for left motors two is for right motors) of SPDT (single pole double throw) relays were used to control the DC motor's rotation direction.

Relays are electrically operated switch used to isolate one electrical circuit from another. A relay consists of a coil used as an electromagnet to open and close switches contacts. As the two circuits are isolated from one another, a lower voltage circuit can be used to operate relay, which will control a separate circuit that requires higher voltage or amperage.

SPDT relay configuration switches one common pole to two other poles flipping between them.

As shown in Figure 3.19, the common point E completes a circuit with C point when the relay coil is at a rest, as no voltage is applied to it. The circuit is closed. When power is applied to the coil, metal level is pulled down, closing the circuit between points E and D, opening the circuit between E and C (Dan, 2012).

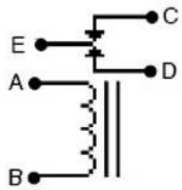


Figure 3.19. SPDT relay schematic diagram

1n5822 Diodes; Diode is an electronic component with two electrodes called as the anode and the cathode. Diodes can be used as rectifiers, signal limiters, voltage regulators, switches, signal modulators and signal mixers. Two pieces of 1n5822 model diodes were used in DC motors control unit as rectifiers. They convert current from the batteries to DC, which flows in only one direction. When the cathode is negatively charged relative to the anode (at a voltage greater than a certain minimum value) current flows through the diode. If the cathode is positive with respect to the anode (at the same voltage as the anode, or negative by an amount less than certain minimum value voltage) the diode does not conduct current. The working principle of the diode (Margaret, 2005) is shown in Figure 3.20

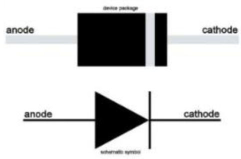


Figure 3.20. Working principle of the diode

IRF 3205 Power Mosfet; Mosfet (metal oxide semiconductor field effect transistor) is a special type of field effect transistor. Mosfet is voltage controlled device. It has gate, drain and source terminals. By applying voltage at the gate, Mosfet generates an electrical field to control the current flow through the channel between drain and source. There is no current flow from the gate into the Mosfet. The terminals of the Mosfet (Anonymous, 2013 j) is shown in Figure 3.21

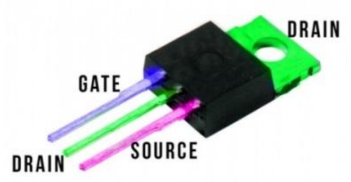


Figure 3.21. Terminals of the Mosfet

Mosfet is variable resistor, where the Gate-Source voltage difference can control the Drain-Source resistance. When there is no applying voltage between the Gate-Source, the Drain-Source resistance is very high, which is almost like an open circuit so current can't flow through the Drain-Source. When Gate-Source potential difference is applied, the Drain-Source resistance is reduced and there is current flowing through Drain-Source, which will be a closed circuit.

Two pieces of N-Channel type of Mosfet was used in DC motors control unit to control the motor rotation speed. For N channel type of Mosfet, the source terminal is connected to ground. To turn the Mosfet on, raised voltage on the gate is required.

Capacitors; Capacitor is a little like a battery. Although, capacitor and battery work in completely different ways, they both store electrical energy.

When a capacitor is connected to the battery; once it's charged, the capacitor has the same voltage as the battery. For a small capacitor, the capacity is small but large capacitors can hold quite a bit of charge.

A capacitor's storage potential called as capacitance is measured in unit's farads. 1 farad capacitor can store one coulomb of charge at 1 volt. One ampere represents a rate of electron flow of 1 coulomb of electrons per second, so 1 farad capacitor can hold 1 ampere second of electrons at 1 volt. Capacitors are typically measured in microfarads (Marshall, 2010).

Two pieces of 1000 micro farad 25 volt capacitors were used in DC motors control unit to store electrical charge for high-speed use. Capacitors with different capacitance and voltage values were used in other units of the circuits to regulate the charge for microcontrollers and other electronic components. (One 100 microfarad 25 V and one 10 microfarad 35V capacitors are for DC motors control circuit, data acquisition and communication circuit and servo motor control circuit, four 1 microfarad 63V capacitors are for RS 232 communication module). These capacitors can eliminate ripples and spikes come from power supply and then, convert the voltage to almost direct voltage (DC) without ripples.

ULN 2003 Darlington Transistor Array (Relay driver); The ULN 2003 is an IC (integrated circuit) that consists of seven Darlington Transistor pairs with high voltage and current capability. Each channel is rated at 500 mA and can withstand peak currents of 600mA. It includes suppression diodes for inductive load driving. Inputs are pinned opposite the output to simply board layout.

ULN 2003 allows driving high current loads like relays and motors which require more power than microcontroller can supply. 5V voltage on input pin will turn on Darlington Pair Transistor. The load goes between the output pin and the load supply voltage which can be up to 50V. With this feature; 12V relays, electric motors, stepper motors can be driven.

One ULN 2003 Darlington Transistor Array was used in DC motors control unit as relay driver. The schematic diagram of ULN 2003 Darlington Transistor Array (Ligo, 2012) is shown in Figure 3.22.

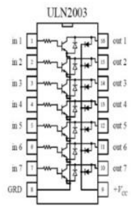


Figure 3.22. ULN 2003 schematic diagram

3.6.2. Data Acquisition and Communication Unit

Data acquisition and communication unit was designed to get data from fire detection unit and to provide communication between microcontroller and netbook. The electronic components of the data acquisition and communication unit are listed below;

1. L7805 Voltage regulator

2. Max 232 integrated circuit

LM7805 Voltage Regulators; LM7805 is a voltage regulator integrated circuit. The voltage source in a circuit may have fluctuations and would not give the fixed voltage output. The voltage regulator maintains the output voltage at a constant value. 7805 provides 5V regulated power supply. The pin diagram of the LM7805 (Anonymous, 2013 e) is shown in Figure 3.23.

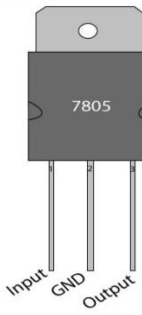


Figure 3.23. Pin diagram of LM7805

Three pieces of LM7805 were used to design and build regulation circuits for data acquisition and communication circuit, DC motors control circuit and servo motor control circuit. The regulation circuits were arranged using one LM 7805, one 100 microfarad 25V and one 10 microfarad 35V capacitors. This circuit was used to filter the spikes and ripple comes from power supply by blocking them and maintain the voltage at 5V DC which is required for microcontroller and other components.

Max 232 Integrated Circuit; The Max 232 IC is used to convert the microcontroller logic levels to RS 232 logic levels during serial communication of microcontrollers with pc. The microcontroller operates at TTL (transistor-transistor logic) logic level (0-5V) while the serial communication in pc works on RS 232 standards (-25V to 25V). This makes it difficult to establish a direct link between computer and microcontroller for communication. In data acquisition and communication unit, a max 232 IC was used to provide an intermediate link between pc and microcontroller. It is a dual driver/receiver that includes a voltage generator to supply RS232 voltage levels from a single 5V supply. The pin diagram of the Max 232 (Anonymous, 2011 b) is shown in Figure 3.24.

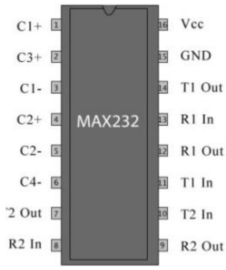


Figure 3.24. Pin diagram of the Max 232

As shown in Figure 3.24 pin 1, 2, 3, 4, 5 and 6 are capacitor connection pins. Each receiver (R1 and R2) converts RS232 inputs to 5V microcontroller logic levels. The drivers (T1 and T2) also called as transmitters; convert the microcontroller input level into RS232 level.

One Max 232 IC and a 9 pins serial port connector were used to establish a serial communication between microcontroller and netbook. The communication process is as below;

1. Receiver pin of the microcontroller was connected to the receiver 1 out pin of the Max 232 IC and third pin of the 9 pin serial connector was connected to receiver 1 in pin of the Max 232 IC to receive information from netbook.

2. Transmitter pin of the microcontroller was connected to the transmitter 1 in pin of the Max 232 IC and second pin of the 9 pin serial connector was connected to transmitter 1 out pin of the Max 232 IC to transmit information to netbook.

3. 9 pin serial connector was connected to netbook using a RS 232 to USB (Universal Serial Bus) converter.

3.6.3. Servo Motor Control Unit

As our servo motor can operate at 5V voltage level, we drove servo motor directly with microcontroller and regulation circuit. Regulation circuit was used to provide 5V regulated power supply and microcontroller was dedicated to produce operating signal to the servo motor.

3.6.4. Microcontrollers

Microcontrollers are special purpose computers. They are embedded inside some other device so they can control the features or actions of the device. Microcontrollers are dedicated to one task and run one specific program. The program can be stored in ROM (read-only memory) or Flash memory. They are often low-power devices. A battery-operated microcontroller may consume 50 miliwatts.

Three Atmega 32 model 8 bit microcontrollers (first is for DC motors control, second is for data acquisition and communication and third is for servo motor control) produced by Atmega Company were used in data acquisition and control circuits. They execute the motion plan, control the motion and obstacle avoidance system through motors and ultrasonic sensors, get data from fire detection unit and transmit them to the netbook. The features of the used microcontroller are listed below (Anonymous, 2012 f);

Speed grade	0-16 MHz
Power consumption	1.1 mA power consumption at active mode
Operating voltage	4.5V-5.5V operating voltage
Input output channel	32 programmable I/O channel
Oscillator	Internal calibrated oscillator
Timer/counter	Two pieces of 8-bit timer/counter and one 16-bit timer/counter
PWM	Four Pulse width modulation channels
ADC	Eight 10 bit analog to digital converter channel
Write/erase cycle	10,000 Flash, 100,000 EEPROM (electronically erasable read only memory), write/erase cycle
Memories	32 Kbytes flash memory, 1024 bytes EEPROM 2 Kbyte RAM
USART	One programmable universal asynchronous receiver/transmitter

The pin configuration of Atmega 32 is shown in Figure 3.25.

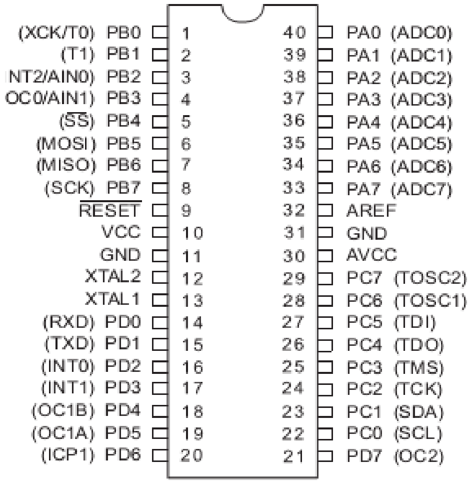


Figure 3.25. Pin configuration of Atmega 32

To operate the Atmega 32 microcontroller;

1. A 4.7 kΩ resistor was connected between 5V power supply and reset pin of the microcontroller.

2. VCC (digital supply voltage) pin of the Atmega 32 was connected to 5V power supply.

3. GND pin of the Atmega 32 was connected to negative terminal of the circuit.

4. AVCC (supply voltage for analog to digital converter) pin of the Atmega 32 was connected to 5V power supply.

5. AREF (analog reference) pin of the Atmega 32 was connected to the 5V power supply.

Operational port and pin configuration of the used microcontrollers are given in Table 3.7. Microcontroller codes are given in Appendix-1.

Table 3.7. Port and pin configuration of microcontrollers

1. microcontroller (DC Motors and obstacle avoidance system control)								
Port Name	**Pin Number**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C	Right motors output signal	Left motors output signal	X	X	Left ultrasound echo signal	Left ultrasound trigger signal	Right ultrasound echo signal	Right ultrasound trigger signal

2. microcontroller (Data acquisition and communication)								
Port Name	**Pin Number**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
A	X	X	X	X	X	ADC Flame sensor input	ADC Smoke sensor input	ADC Temperature sensor input
D	X	X	X	X	X	X	Transmitter of serial communication	Receiver of serial communication

3. microcontroller (Servo motor control)								
Port Name	**Pin Number**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
D	X	Servo motor output signal	X	X	X	X	X	X

3.7. Fire Detection Robot System Architecture

Fire detection robot system contains Motion Planning Unit, Data Acquisition and Control Unit, Obstacle Avoidance Unit, Fire Detection Unit and DC Motor Driver Unit.

Motion planning task of the robot is executed by re-programmable microcontroller. First, the route plan which the robot will follow is programmed to microcontroller. The path distances are defined to robot through motion planning program. After defining process, robot follows these lines so that it executes patrolling task.

Data acquisition and control unit operates like a bridge between the netbook and other units. It directly controls the obstacle avoidance unit. Data acquisition and control unit also responsible to get information from fire detection unit. It gets analog outputs from flame, smoke and temperature sensors then converts them to digital data via ADC channel of the microcontroller. After collecting the information about the environmental situation, data acquisition and control unit transmits fire information to the netbook.

The obstacle avoidance unit was produced using two ultrasonic sensors. They have 15˚ measuring angle. Ultrasonic sensors need at least 10 µs pulse trigger signal for operation. This trigger signal is provided by data acquisition and control unit. With trigger signal, they send eight 40 kHz sound wave. After trigger signal they get echo information and then, send data to control unit that whether there is an obstacle or not. Control unit checks the situation. If there is an obstacle and the distance from obstacle is smaller than threshold value, robot will start to search for free path. When robot finds the free path, it changes direction and goes towards the free path.

Fire detection unit consists of smoke, temperature and flame sensors. Flame sensor was mounted on a servo motor. While robot is executing its patrolling task, flame sensor scans the environment for fire information. Flame sensor has 180˚ freedom of movement which is provided by servo motor (90˚ right side and 90˚ left side) for scanning. All of the sensors acquire the environmental situation and each of them sends their own information to the netbook through data acquisition and control unit dynamically.

Motor driver unit is responsible for motor’s actions. The direction of the motor rotation is controlled by motor driver unit. By motor direction control, robot is driven forward or backward. The speed of the motor rotation is also regulated by motor driver with PWM method. As robot is steered with differential method, motor speed regulation is important for robot turnings.

The schematic of the fire detection robot system architecture is shown in Figure 3.26.

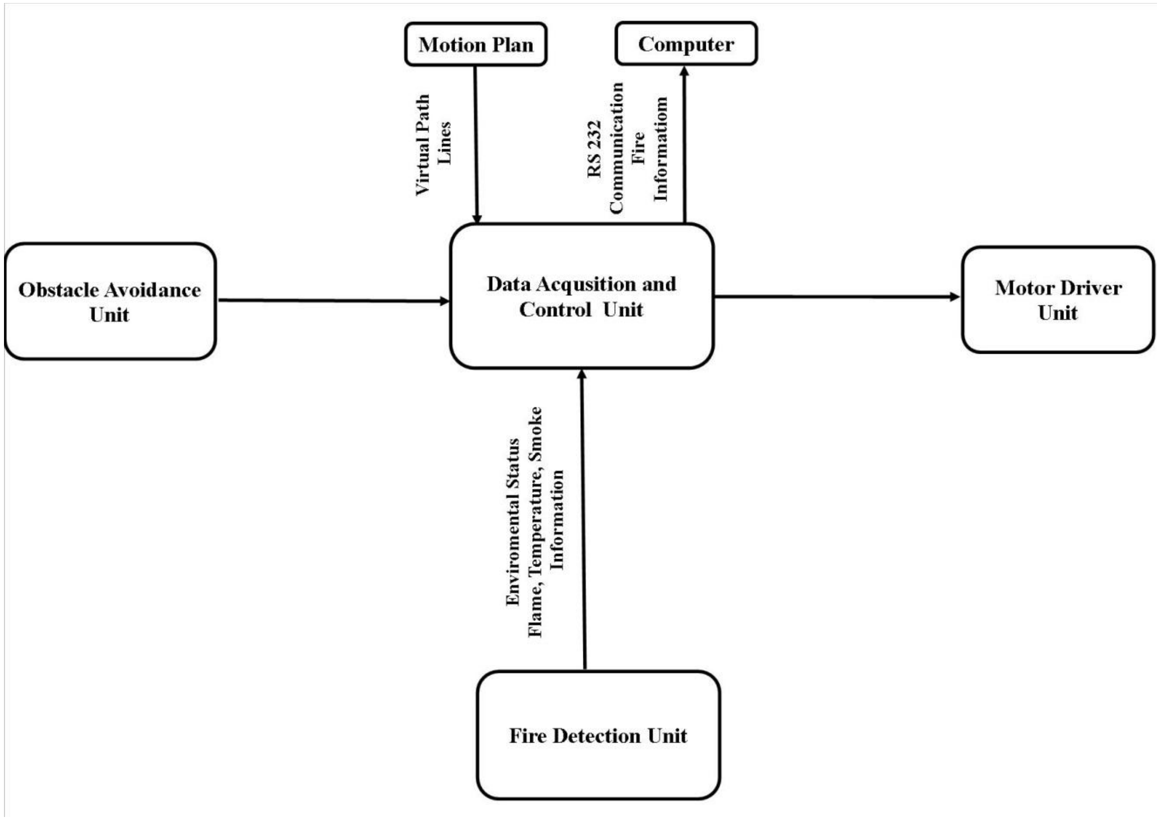


Figure 3.26. System architecture of fire detection robot

3.8. Fire Detection Robot System Functions

3.8.1. Obstacle Avoidance

Obstacle avoidance function of the fire detection robot system consists of two sub functions; first one is obstacle detection and second is measurement of the distance from robot. It can move freely without any collision with any obstacle using obstacle detection and obstacle measurement sub functions so that it can perform patrolling task without any interruption.

As mentioned before obstacle avoidance was implemented using two ultrasonic sensors which mounted to front side of the robot (left ultrasonic and right ultrasonic). There are three basic terms in this function; safe space, danger space and distance (ultrasonic sensor value). Ultrasonic sensors scan for the obstacle occurrence and measure the distance from obstacle dynamically which will keep the robot not too close to any obstacle. With this measurement, data acquisition and control unit decide that the robot is in safe space or danger space using obstacle avoidance algorithm.

Safe distance value is the preset threshold value for the distance between robot and obstacle. Distance value is updated by measurements when robot moves at its patrolling path. After scanning for the obstacle distance, sensors send their echo time values to microcontroller. As indicated at formula 3.17;

$D = t / 58$

The value of safe distance is determined as 60 cm. So the maximum echo time value for safe space is;

Although, the distance calculation formula is given as D = t / 58 in datasheet of the sensors, we formulated the calculation as D = t / 22 with calibration tests.

$t = D \times 22 = 60 \times 22 = 1320$

Fire detection robot moves freely on its patrolling path, when the measured distance value is bigger than the threshold echo time value 1320. If measured echo time value is equal or smaller than the threshold value robot will search for possible free path, first on the right then left side. When it finds the free path whether right or left it changes the direction, goes towards the free side and then goes straight.

As a summary;

1. If distance value (ultrasonic sensor value) > safe distance value;

   Robot will continue the movement.

2. If distance value ≤ safe distance value;

   Robot will change its direction, find the free path.

3.8.2. Operating the Ultrasound Sensors

For obstacle avoidance, we programmed the microcontroller to operate the ultrasound sensors as desired.

These sensors need a trigger to start the measurement process and then they send out an ultrasound at 40 kHz and receive its echo. Sensors calculate distance counting the time between sending and receiving. The steps used to operate the ultrasound sensors are given below;

1. Right ultrasound sensor's trigger and echo pins are respectively connected to zeroth and first pins of C port of the microcontroller.

2. Left ultrasound sensor's trigger and echo pins are respectively connected to second and third pins of C port of the microcontroller.

3. Trigger pins of the sensors are supplied with 15 μs pulses and then they are set to zero (no supply) and waited for 100 μs.

4. The echo pin values of the sensors are ignored when its output is equal to zero and time up to 2,000 μs to wait the settling time of the sensors.

5. When echo pins receive signal which means there is obstacle, they start to count the value of time interval. This value is the time between the sending and returning of the ultrasound. The maximum counting value is set as 60,000 μs.

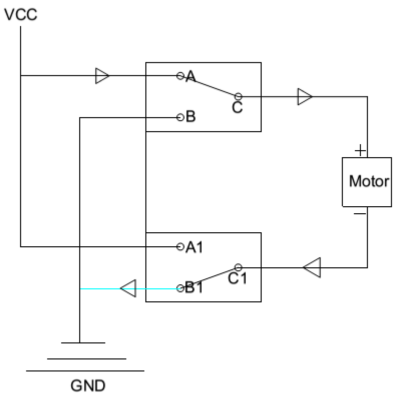
6. The distance of the obstacle is calculated by microcontroller dividing the time value with coefficient 22.

3.8.3. Motion Control

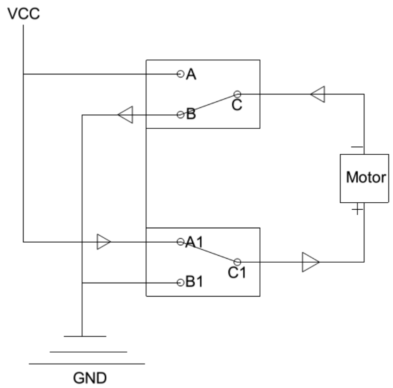
Motion controlling is an important function of the robot. In the operation environment, it needs motion control to execute path tracking and motion planning and obstacle avoidance tasks. For path tracking robot has to act with different movements such as going straight, coming back and turning. When a strange object is detected, robot has to change its direction to avoid obstacle. To carry out these tasks an intelligent motion control is required for fire detection robot system.

In the system DC brushed motors were used. The drive and motion control is convenient to operate. As long as both ends of the motor load within the rated voltage, the motion of the motors can be controlled.

The rotation directions of the motors were controlled with four relays. Two of them were used for left motors and the other two were for right motors. The principle of the motor control with relays is shown in Figure 3.27.



(a)



(b)

Figure 3.27. Rotation direction of the motors (a) Clockwise rotation (b) Counter clockwise rotation

To drive robot motor forward; motors are rotated clockwise direction. As seen in Figure 3.27-a; A pins of the relays are connected to VCC (5V) and B pins are connected to ground. For clockwise rotation; close circuits are completed between A and C in the first relay and between B1 and C1 in the second relay .To drive robot backward; motors are rotated the counter clock wise rotation. Close circuits sides are changed as seen in Figure 3.27-b. Close circuits are completed between B and C in first relay, between A1 and C1 in second relay. Electrical current flow direction is changed for two situations to change the direction of the motor rotation (First situation is for clockwise rotation and second is for counter-clockwise rotation). Four relays are controlled with ULN 2003 relay driver in the system.

As the system is driven with differential drive steering method, speed control is required for turnings. Robot is able change the motion direction using PWM signal and direction control. The motion conditions are indicated below;

1. Four wheels are moving at the same speed, in the same direction, robot goes straight.

2. Four wheels are moving in the same direction but left wheels are moving faster than right wheels, robot turns right at a certain radius.

3. Four wheels are moving in the same direction but right wheels are moving faster than left wheels, robot turns left at a certain radius.

4. Four wheels rotate at the same speed rate but in opposite direction, robot spins in the place.

The relationship between the direction-speed control and robot movement is shown in Table 3.8;

Table 3.8. Direction-speed and motion control

Direction		**PWM signal**		**Direction of wheels**		**Movement of the robot**
Left	Right	Left	Right	Left	Right	
Forward	Forward	100	100	Forward	Forward	Forward
Forward	Forward	60	100	Forward	Forward	Turn left at a certain radius
Forward	Forward	100	60	Forward	Forward	Turn right at a certain radius
Forward	Forward	0	100	Stop	Forward	Turn left sharply
Forward	Forward	100	0	Forward	Stop	Turn right sharply
Backward	Forward	100	100	Backward	Forward	Spinning
Backward	Backward	100	100	Backward	Backward	Backward

A pair of IRF 3205 power Mosfet was used in the system to control the rotation speed of two motors. The working principle of motion speed control with Mosfet is shown in Figure 3.28;

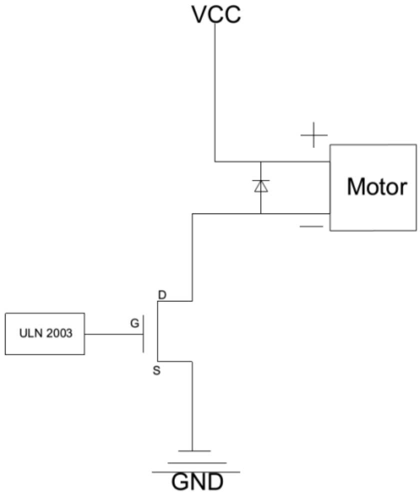


Figure 3.28. Speed control with power Mosfet transistor

As shown in Figure 3.28 source pin of the Mosfet is connected to the ground, gate pin is connected to ULN 2003 driver and drain pin is connected to motor’s load end. When power Mosfet gate pin is acquired an operating signal from ULN 2003 it completes close circuit between drain and source pins so that motor works with maximum power and speed. When ULN 2003 cuts off the operating signal, power Mosfet converts the close circuit to open circuit. Electrical current starts to flow between end loads of the motor in a short way which is completed with 1n5822 diode. In this situation; motor starts to use its inertial load and energy for operating. As the energy is reduced in time, motor operates slower than its maximum value. The time interval between the mosfet’s open circuit and close circuit designate the duty cycle of the motors. The principle of PWM method is shown in Figure 3.29.

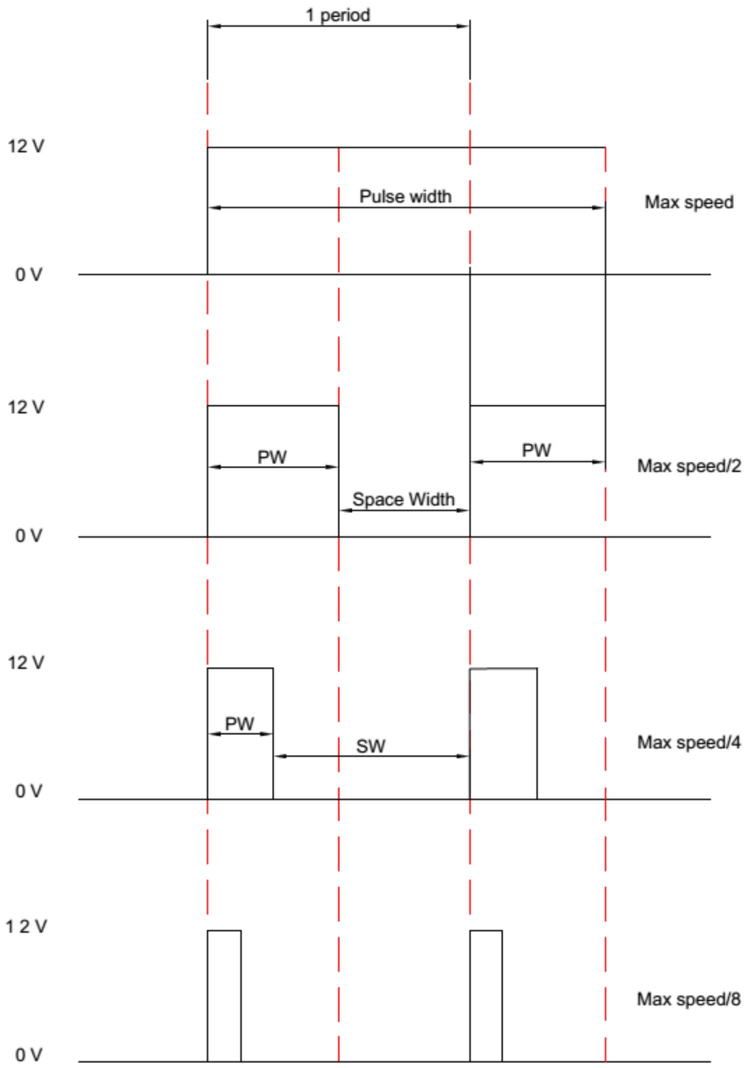


Figure 3.29. PWM method

3.8.4. Fire Detection

Three types of sensors were used in the system to detect fire event. With the varieties of the sensor (smoke, temperature and flame) it is possible to get more reliable and high accuracy results from the fire detection function. A sensor data fusion was used to obtain high reliable results from the fire detection unit.

In fire detection unit, the weight is a function of the P_D (probability of detection) of the sensors. Although the probability values may vary with time in real life, the optimum detection probability values can be determined for smoke, flame and temperature sensors through experiments. The data fusion structure of the fire detection unit is shown in Figure3.30.

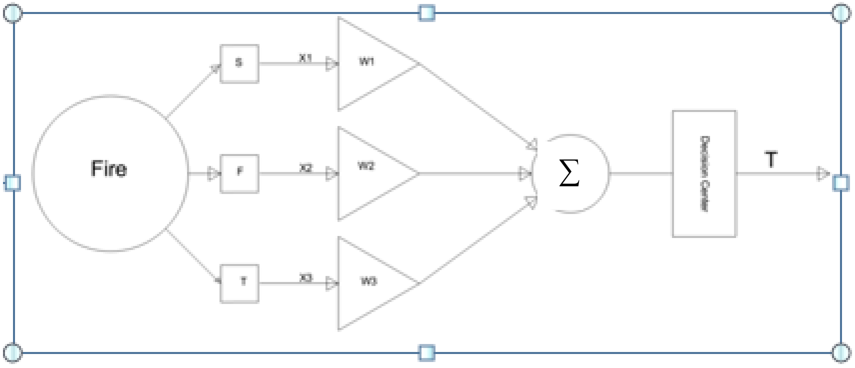


Figure 3.30. Sensor data fusion structure of fire detection unit

In the data fusion system of the fire detection unit the detection values of the each sensor; smoke (X1), flame (X2) and temperature (X3) are acquired by data fusion decision center. These values are considered and multiplied with weight values and final decision T is determined. The formula of the final decision is;

$$T = W_1X_1 + W_2X_2 + W_3X_3 \tag{3.18}$$

T > threshold value, there is a fire.

To find the optimum weight values for each sensor and to get accurate fire detection information; different variations have been experimented for different detection distances and fire sources.

The computation of the rule was implemented in microcontroller of the data acquisition and communication unit. Data acquisition and communication unit received fire information from fire detection unit through and ADC channels of the microcontroller. Then, this information was interpreted and it was decided that there is a fire or not using data fusion algorithm.

To derive the fire detection equation; PLS (Partial Least Square) analysis which is a method of MLR (Multiple Linear Regression) was used. To setup the analysis; number of factor was indicated as 15, Simple PLS algorithm and Split Cross Validation method were used. Analysis model was set using 200 samples for temperature, smoke and flame values. After settings, the analysis model was run using SAS (Statistical Analysis System) software. The result of the analysis is given below;

Calculated number of factor	3
Accuracy of the estimator	%97
Coefficient of regression for temperature	0.16
Coefficient of regression for flame	0.77
Coefficient of regression for smoke	0.07

Using the results of the analysis the fire detection equation was derived as;

Fire = (Temperature x *0.16) + (Flame* x *0.77) + (Smoke* x *0.07)* (3.19)

The flowcharts of the obstacle avoidance and fire detection processes are shown in Figures 3.31 and 3.32 respectively.

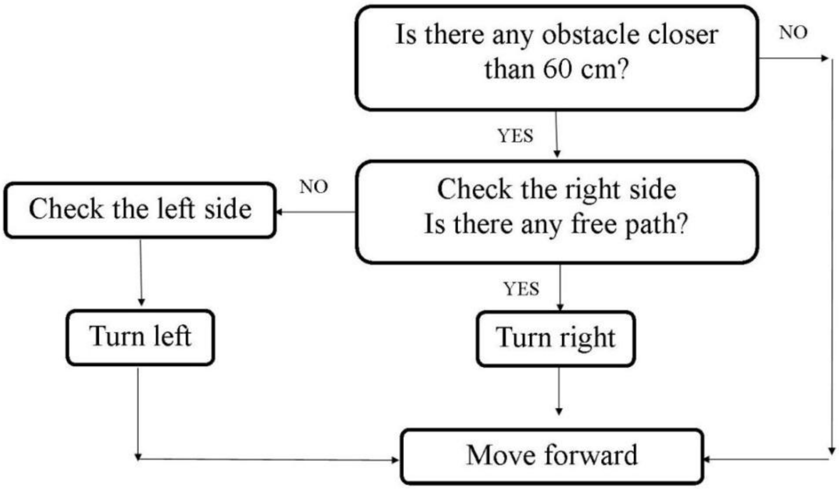


Figure 3.31. Flowchart of the obstacle detection function

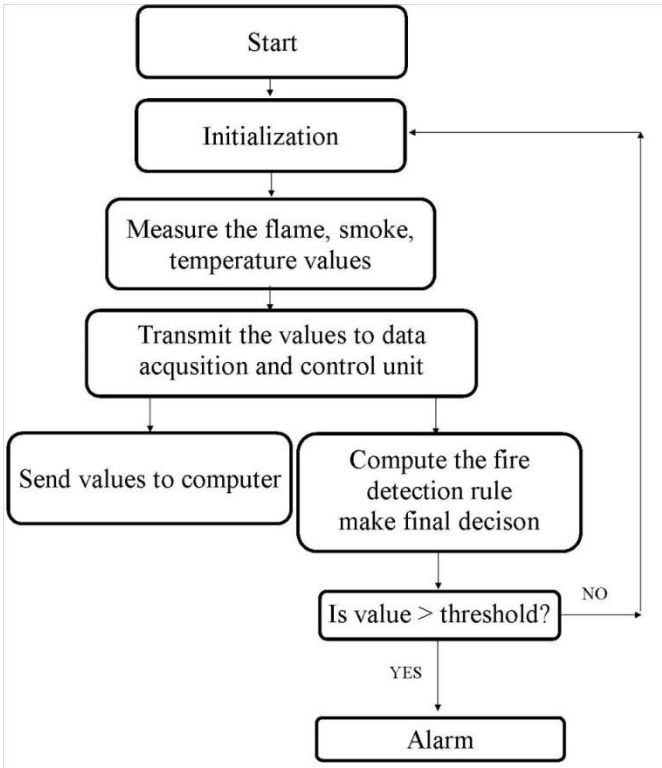


Figure 3.32. Flowchart of the fire detection function

4. RESULTS AND DISCUSSIONS

In this chapter of thesis; the applied function tests of the robot are described and test results are discussed. Tests are categorized into four general parts;

1. Path Tracking with Different Speed Values

2. Implementation and Test of the Obstacle Avoidance Structure

3. Fire Detection Test

4. Sensor Data Fusion Algorithm Test

4.1. Path Tracking Test with Different Speed Values

First test for the fire detection robot was path tracking. As mentioned before; system has ability to move with different speed values. For path tracking test, 3 different speed values were used. These were 0.5 m/s (maximum), 0.3 m/s and 0.2 m/s (minimum). For path tracking test, we established a rectangular test environment with four corner points (A, B, C and D). The distance between A and B points is 425 cm, B and C points is 250 cm, C and D is 425 cm and D and A is 250 cm. The sketch of the test environment is shown in Figure 4.1.

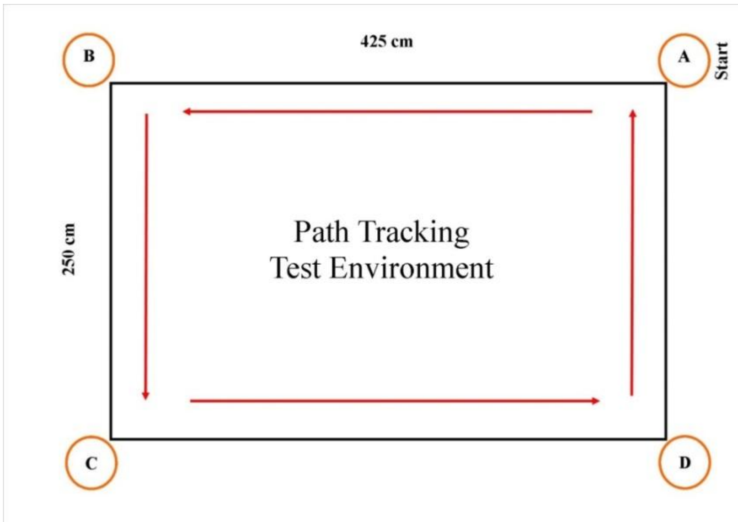


Figure 4.1. Path tracking test environment

A motion plan algorithm was programmed to DC motors control unit. As motion plan; robot starts to operate at A point and try to reach B point. Then, it waits for a short time interval, turns left and moves forward. It executes the same process to reach C, D and finish (A) point.

Path tracking test scenario was applied for 0.5, 0.3 and 0.2 m/s speed values. Wait times were added to algorithm because of the microcontroller's processor working ability. Tracking ability of the robot was observed. The test settings and the results for 0.5, 0.3 and 0.2 m/s speed values are given in Table 4.1.

Table 4.1. Path tracking test settings and results

Path tracking test for 0.5 m/s speed value			
Test Settings	**Value**	**Results**	**Value**
Programmed move forward time for A-B and C-D lines	9,000 ms (millisecond)	Elapsed move forward time for A-B and C-D lines	18 s
Programmed move forward time for B-C and D-A lines	5,000 ms	Elapsed move forward time for B-C and D-A lines	10 s
Programmed left turn time	2,300 ms	Total elapsed time for turns	9.2 s
<u>Total path distance</u>	1,350 cm	<u>Total elapsed time without turnings and waits</u>	28 s
Programmed speed	0.5 m/s	<u>Calculated speed</u>	0.48 m/s
		Total wait time	8 s
		Total elapsed time	45.2 s
Path tracking test for 0.3 m/s speed value			
Test Settings	**Value**	**Results**	**Value**
Programmed move forward time for A-B and C-D lines	15,000 ms	Elapsed move forward time for A-B and C-D lines	30 s
Programmed move forward time for B-C and D-A lines	9,000 ms	Elapsed move forward time for B-C and D-A lines	18 s

Programmed left turn time	2,300 ms	Total elapsed time for turns	9.2 s
Total path distance	1,350 cm	Total elapsed time without turnings and waits	48 s
Programmed speed	0.3 m/s	Calculated speed	0.28 m/s
		Total wait time	8 s
		Total elapsed time	65.2 s
Path tracking test for 0.2 m/s speed value			
Test Settings	**Value**	**Results**	**Value**
Programmed move forward time for A-B and C-D lines	22,500 ms	Elapsed move forward time for A-B and C-D lines	45 s
Programmed move forward time for B-C and D-A lines	13,500 ms	Elapsed move forward time for B-C and D-A lines	27 s
Programmed left turn time	2,350 ms	Total elapsed time for turns	9.2 s
Total path distance	1,350 cm	Total elapsed time without turnings and waits	72 s
Programmed speed	0.2 m/s	Calculated speed	0.19 m/s
		Total wait time	8 s
		Total elapsed time	89.2 s

In this test, it was observed that with the specified move forward and left turn time values; fire detection robot could reach the corner points of the test environment, turned the corners with 90˚ angle sharply and tracked its path lines accurately. But it has to be noted that; as the rolling resistance is changed according to the road surface, road surface can affect the speed, turn angle and total traveled distance values of the robot. Therefore, left turn and move forward time values should be re-designed for new operation environment. As the motion plan unit of the robot is re-programmable, it can be programmed for any new environmental condition. Sample screenshots from the tests are shown in Figures 4.2, 4.3 and 4.3.

Figure 4.2. Path tracking starting at A point

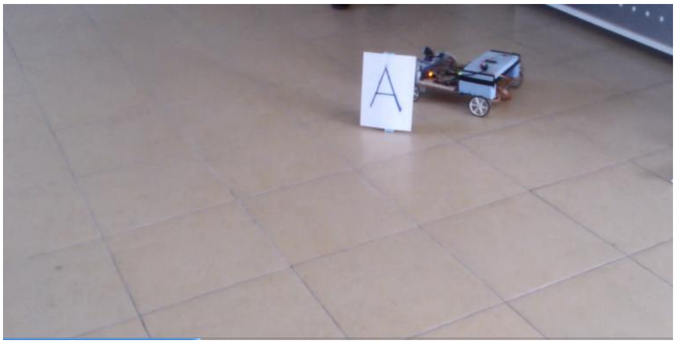


Figure 4.3. Left turn at A point

Figure 4.4. Path tracking between B and C points

4.2. Obstacle Avoidance

Second function test of the fire detection robot was obstacle avoidance. We designed and implemented the obstacle avoidance function of the system based on two ultrasound sensors (right and left).

4.2.1. Implementation and Test of the Obstacle Avoidance Structure

In the obstacle avoidance structure implementation process; the sensors were mounted to carrier board with 15° measurement angle to avoid the interference between the sounds. After placement, a basic algorithm for obstacle avoidance was developed. The details of this algorithm are given below;

1. If the measured distance value of right or left ultrasound sensor is smaller than 60 cm, robot will start to search for free place.

2. If the measured distance value of right ultrasound sensor is bigger than measured distance value of left ultrasound sensor which means free place is closer to right side, right motors will stop and left motors will run for 60 ms. System repeats the same process and turns right until it reaches the free place and then moves forward.

3. If the measured distance value of left ultrasound sensor is bigger than measured distance value of right ultrasound sensor which means free place is closer to left side, left motors will stop and right motors will run for 60 ms. Robot turns left until it reaches the free place and then moves forward.

For the obstacle avoidance function test, an environment which contains five obstacles was established.

In obstacle avoidance test it was observed that the obstacle avoidance structure and algorithm of the system could satisfy requirements of the fire detection robot. The performance of the robot was tested for five times in the established test environment. It could reach finish point without any collision or interruption. Sample screenshots from the tests are shown in Figures 4.5, 4.6 and 4.7.

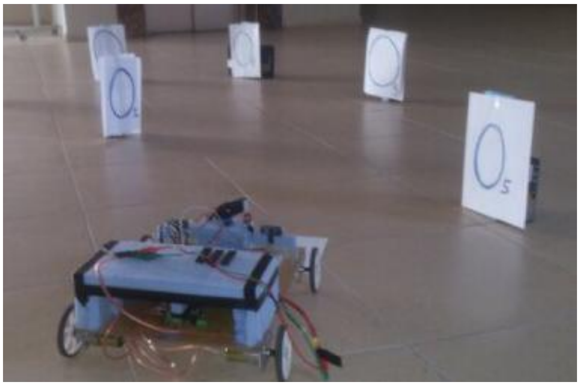


Figure 4.5. Obstacle avoidance test environment

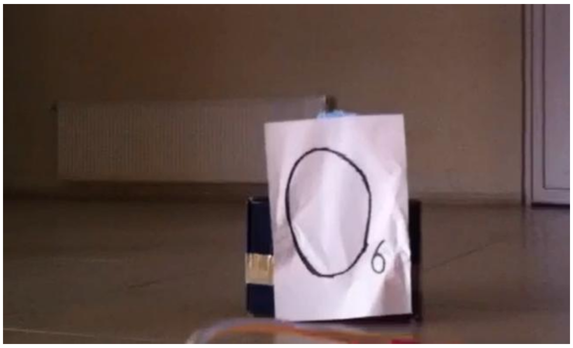


Figure 4.6. Avoidence of sixth obstacle (80 cm away from the obstacle)

Figure 4.7. Avoidance of sixth obstacle (60 cm away from the obstacle)

4.3. Fire Detection Test

The fire detection capability of the system was tested in this part using candle and firewood as fire sources under controlled indoor and outdoor conditions. In order to evaluate fire detection capability of the robot, two dimensions were tested. First one was the distance of the robot from the fire sources. Three different distances were taken into consideration based on two different fire sources mentioned above. The second dimension was concerning about the capability of the robot while it was tracking its path around the fire sources.

To execute the fire detection capability tests; a netbook was mounted to second layer of the carrier board and a serial connection was established between the data acquisition unit and netbook to receive and save the measurement data of the sensors. A sample picture of this configuration is shown in Figure 4.8.

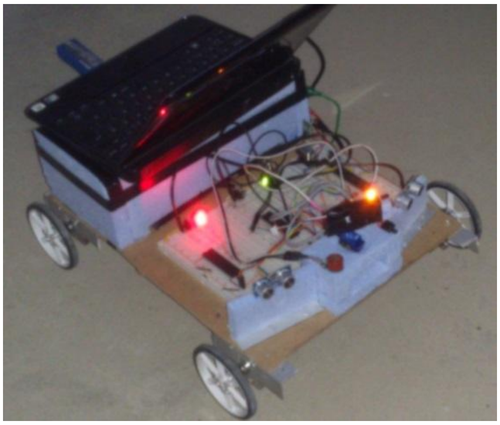


Figure 4.8. Robot configuration with netbook

4.3.1. Detection Test Based on Distances

In the distance test, robot's reaction to certain test conditions according to three different reference points away from the fire sources were taken into consideration. These test conditions are non-fire environment and fire within the range.

The aim of these tests was to measure the reaction of the sensors while they are located at 50 cm, 75cm and 100 cm distances from the fire source. The detection tests were executed with the candle and firewood.

At first step of the test; the intense of IR light, smoke and temperature values of the non-fire environment were measured for 60 seconds via sensors and saved to netbook. At the next step, robot was located at 50, 75 and 100 cm distances from candle flame respectively. The variation of IR light, smoke and temperature values were measured and saved for 140 seconds. The same process was applied for firewood flame. Relevant obtained values based on different distances are summarized in Figures 4.9 to 4.26.

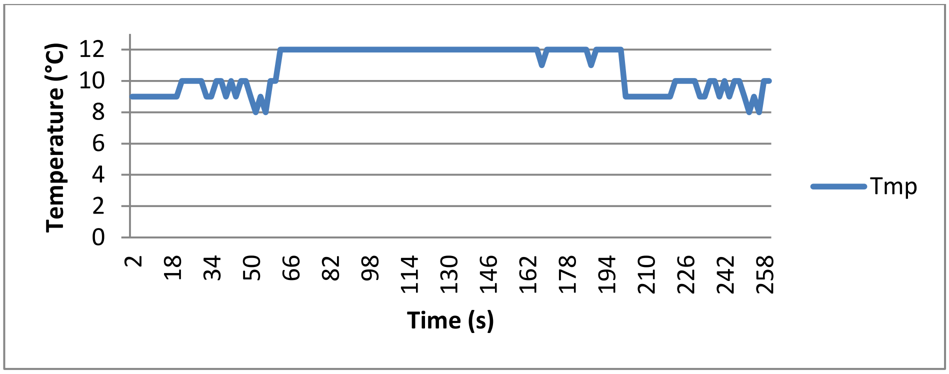


Figure 4.9. Temperature sensor values-50 cm from the candle flame

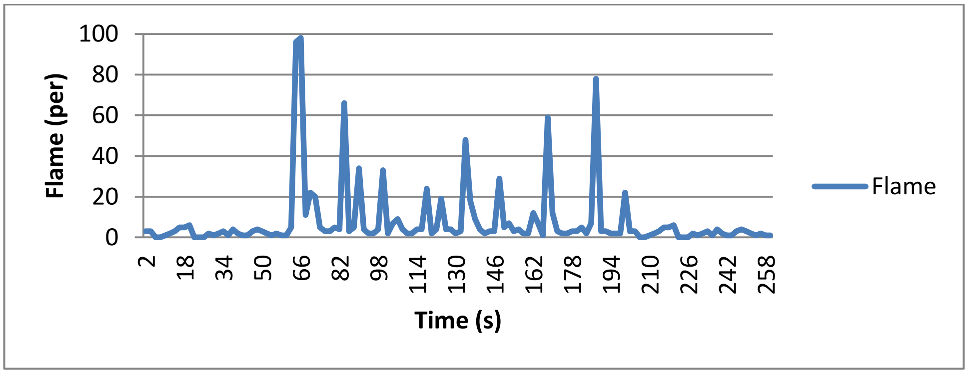


Figure 4.10. Flame sensor values-50 cm from the candle flame

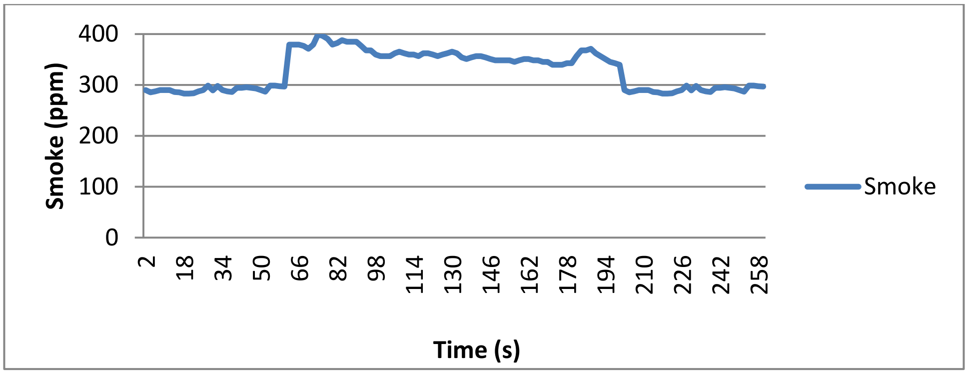


Figure 4.11. Smoke sensor values-50 cm from candle flame

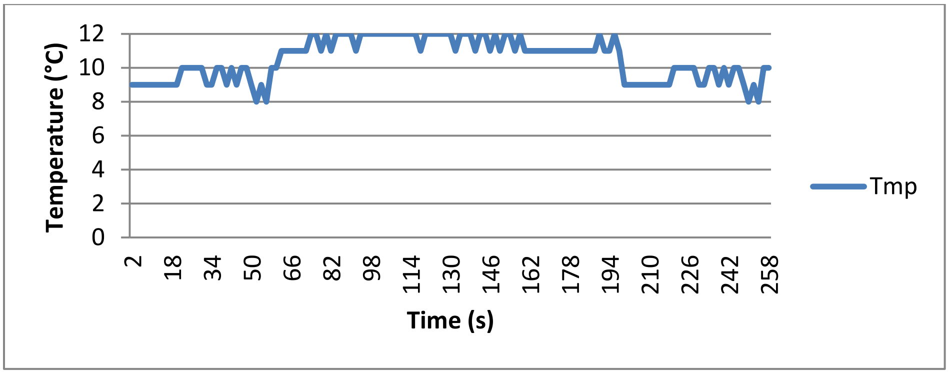


Figure 4.12. Temperature sensor values-75 cm from candle flame

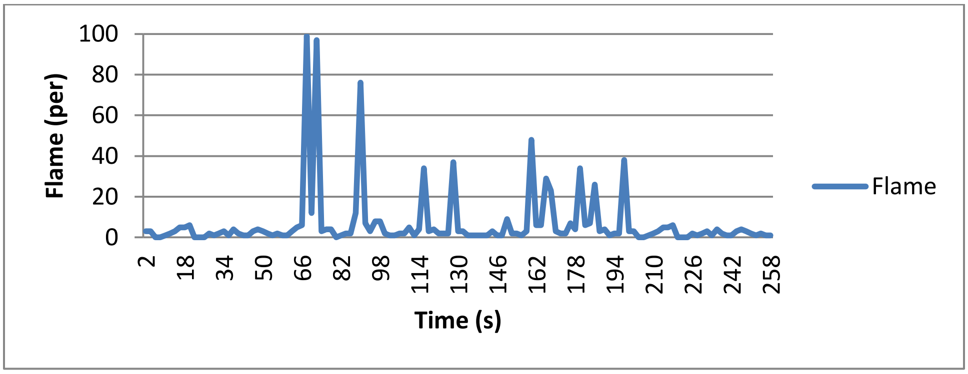


Figure 4.13. Flame sensor values-75 cm from candle flame

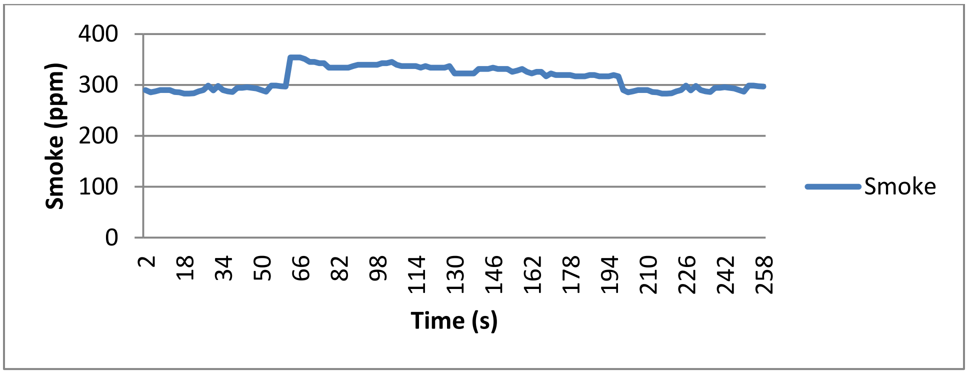


Figure 4.14. Smoke sensor values-75 cm from candle flame

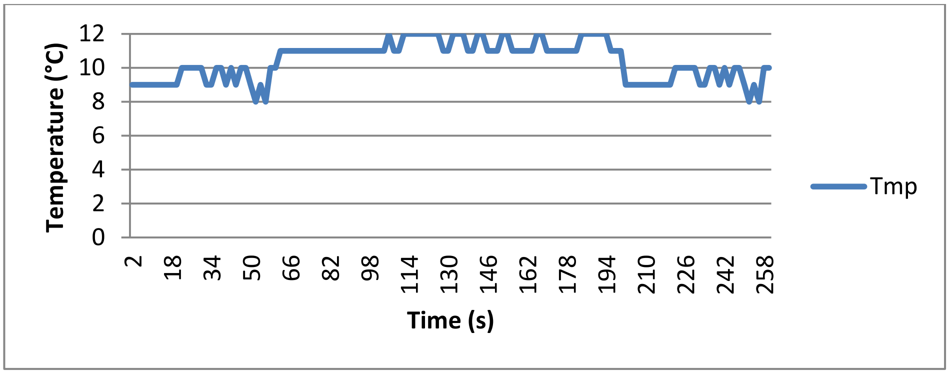


Figure 4.15. Temperature sensor values-100 cm from candle flame

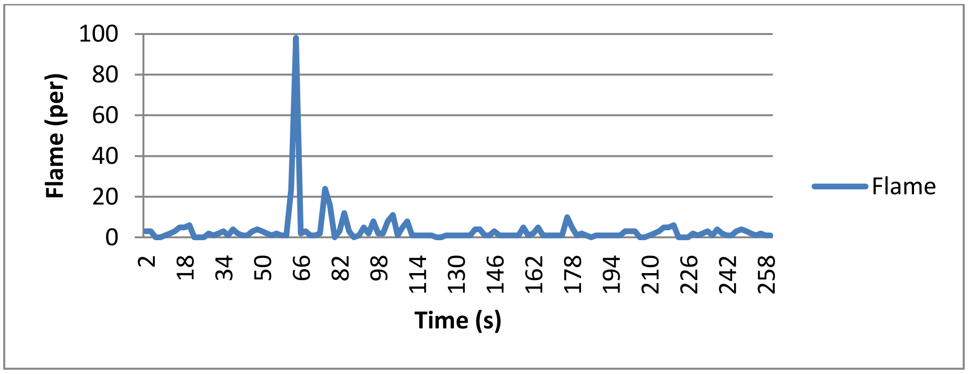


Figure 4.16. Flame sensor values-100 cm from candle flame

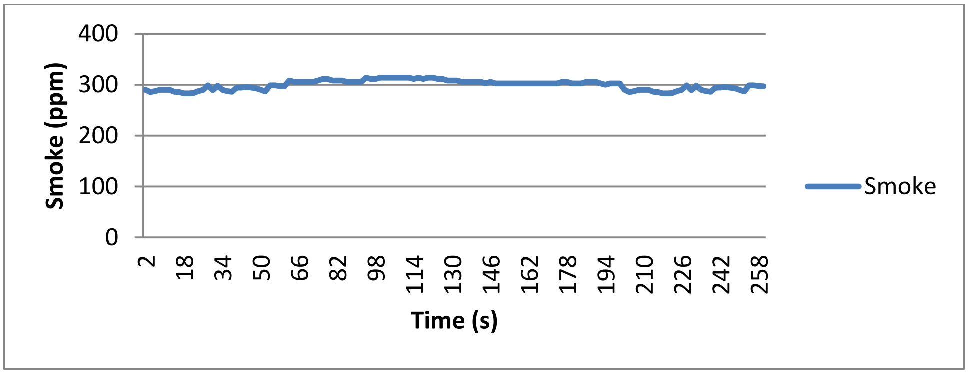


Figure 4.17. Smoke sensor values-100 cm from candle flame

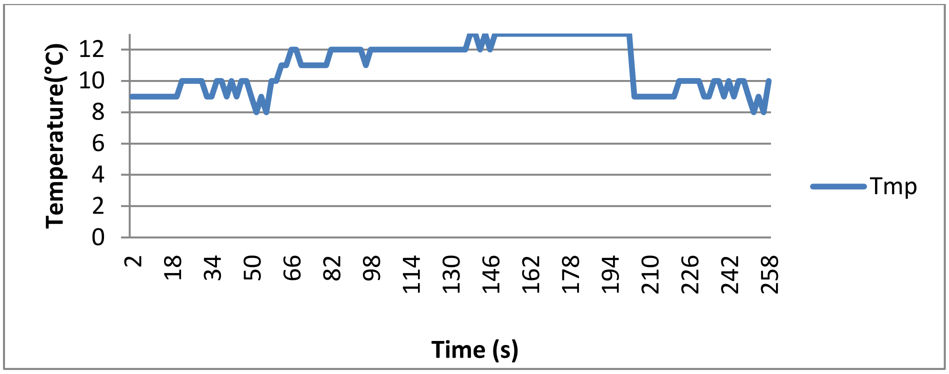


Figure 4.18. Temperature sensor values-50 cm from firewood flame

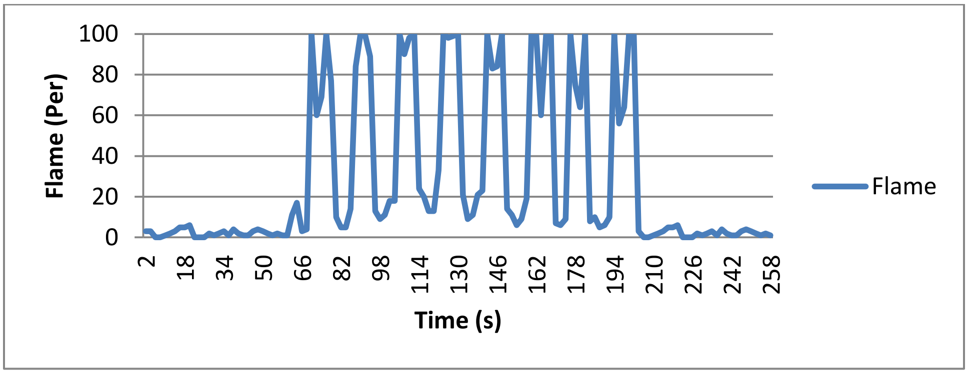


Figure 4.19. Flame sensor values-50 cm from firewood flame

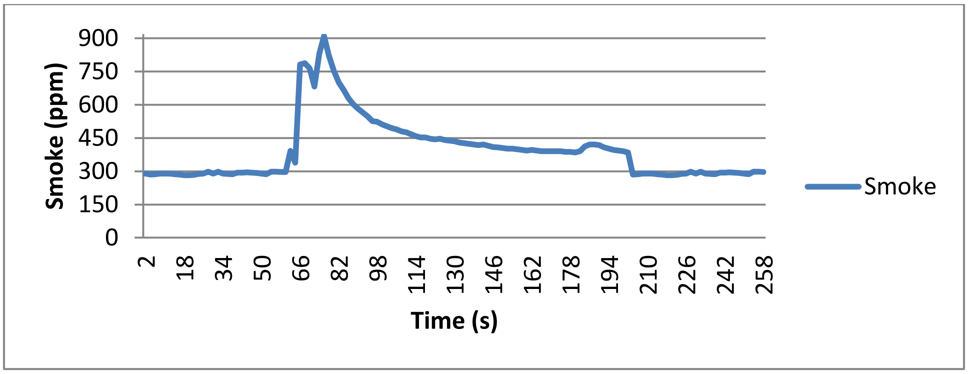


Figure 4.20. Smoke sensor values-50 cm from firewood flame

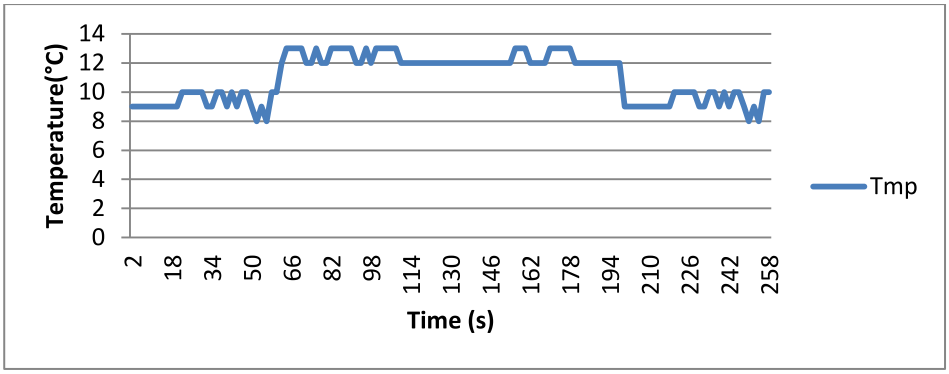


Figure 4.21. Temperature sensor values-75 cm from firewood flame

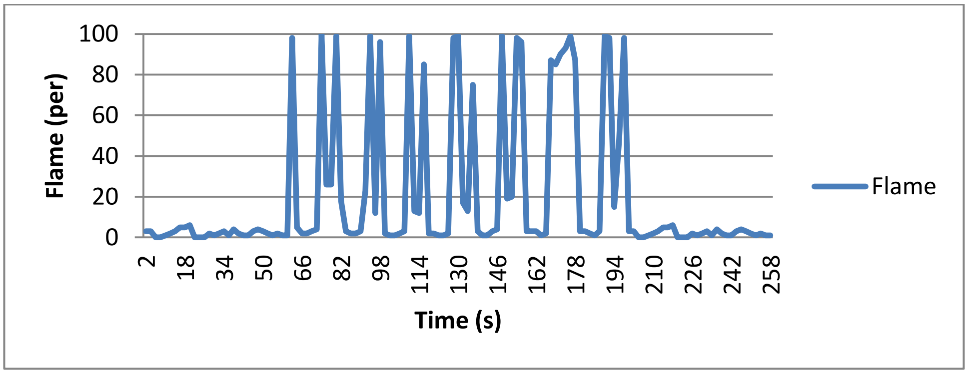


Figure 4.22. Flame sensor values-75 cm from firewood flame

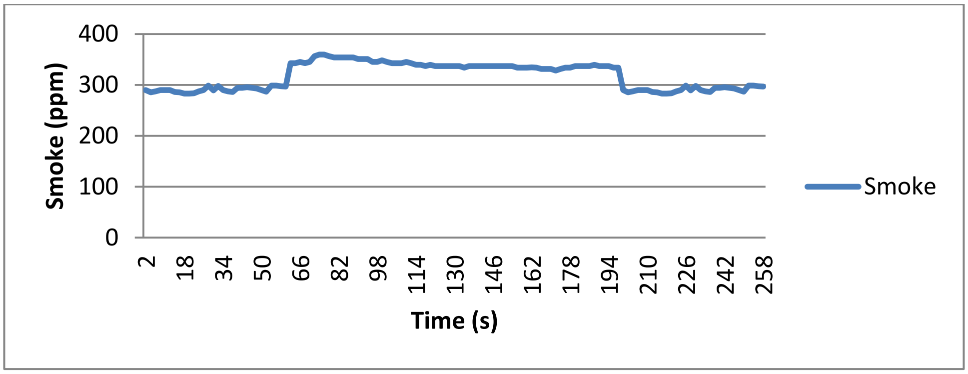


Figure 4.23. Smoke sensor values-75 cm from firewood flame

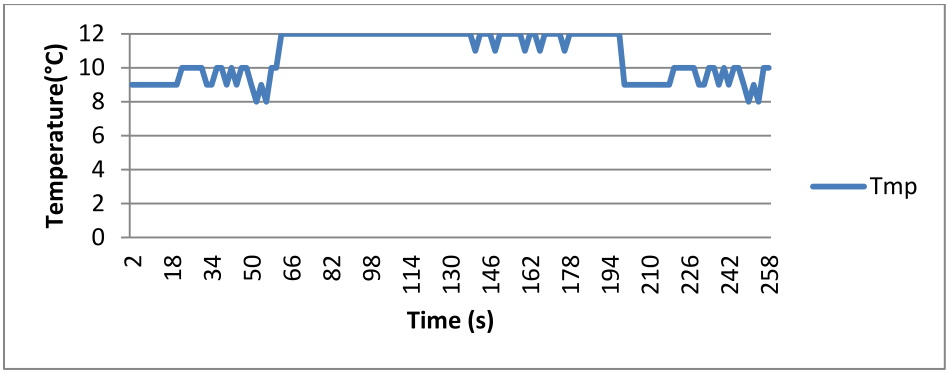


Figure 4.24. Temperature sensor values-100 cm from firewood flame

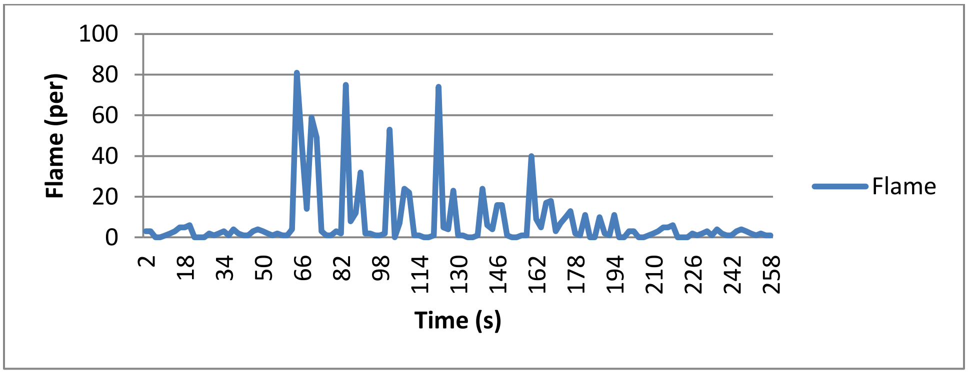


Figure 4.25. Flame sensor values-100 cm from firewood flame

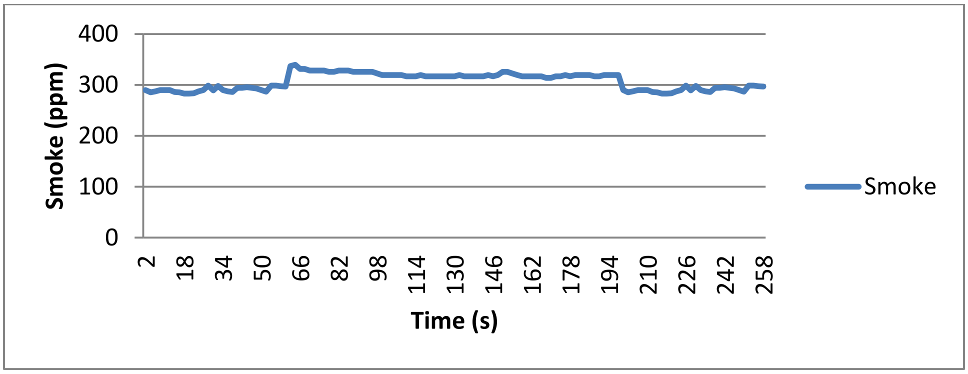


Figure 4.26. Smoke sensor values-100 cm from firewood flame

Using the measurements of based on distances, comparison graphics for candle and firewood flames were derived. These comparison graphics are shown in Figures 4.27 to 4.32.

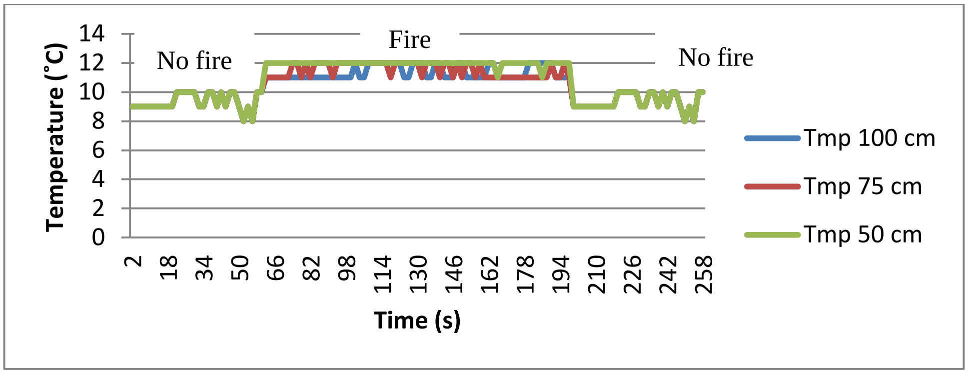


Figure 4.27. Temperature detection comparison of candle flame

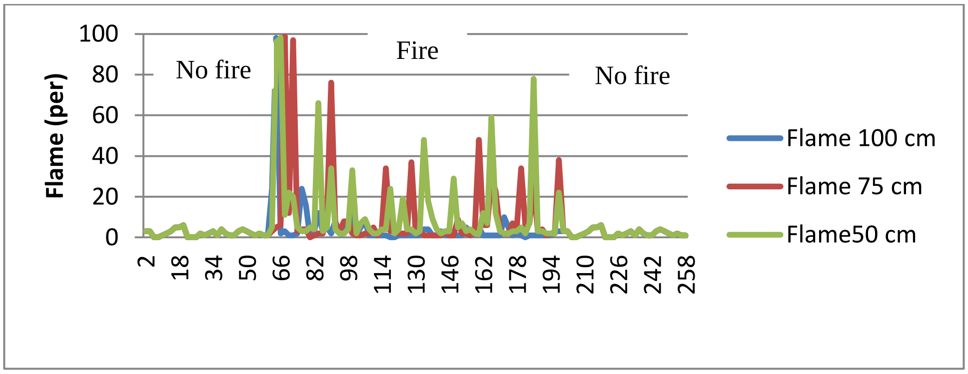


Figure 4.28. Flame detection comparison of candle flame

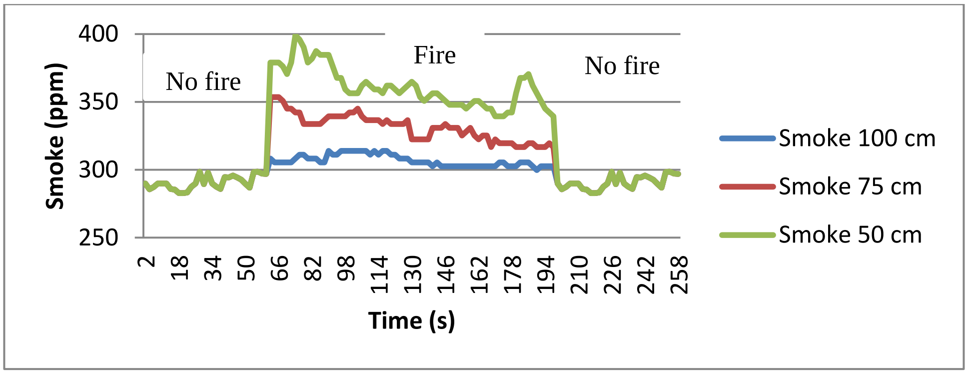


Figure 4.29. Smoke detection comparison of candle flame

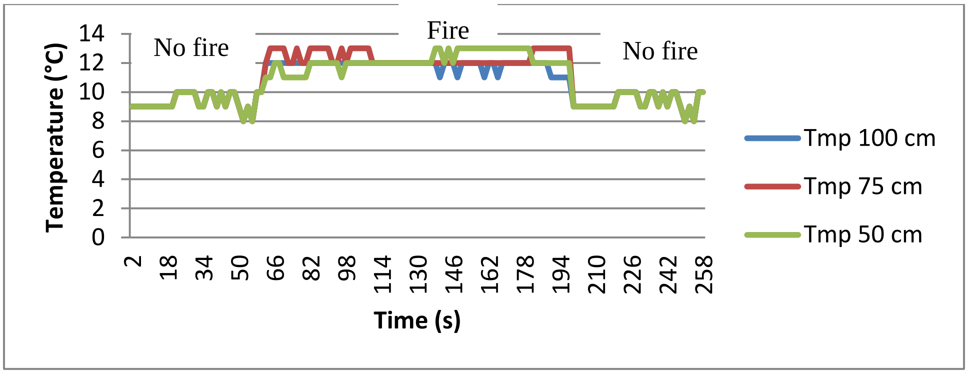


Figure 4.30. Temperature detection comparison of firewood flame

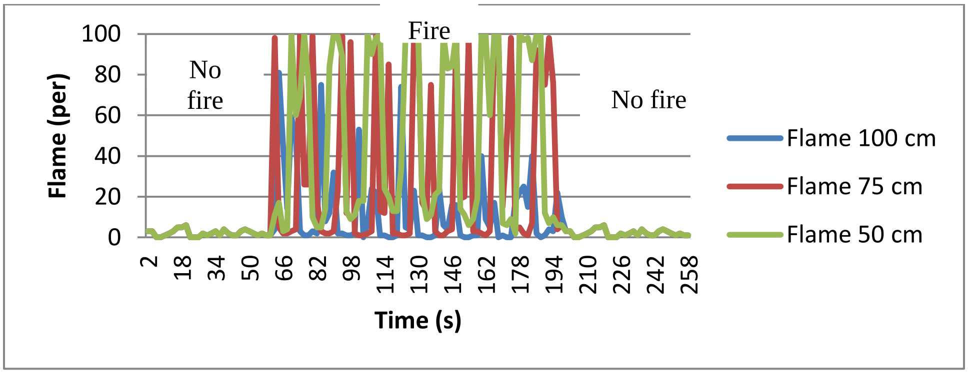


Figure 4.31. Flame detection comparison of firewood flame

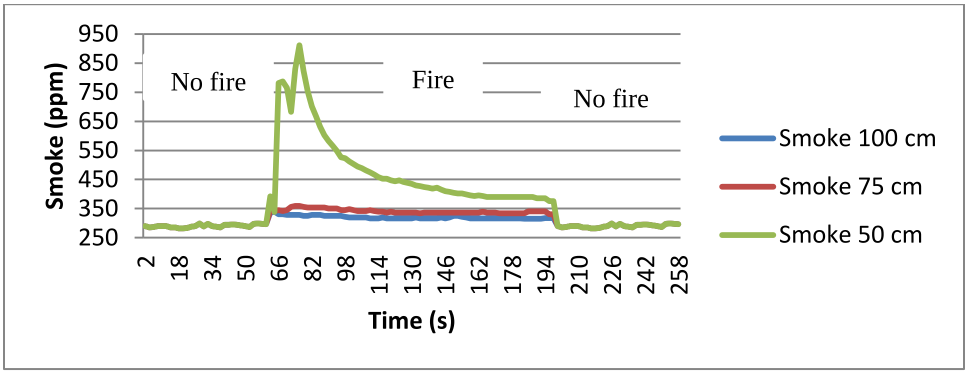


Figure 4.32. Smoke detection comparison of firewood flame

Temperature, flame and smoke measurement values of the candle and firewood flames were compared for 50 cm distance. Comparison graphics are shown in Figures 4.33, 4.34 and 4.35.

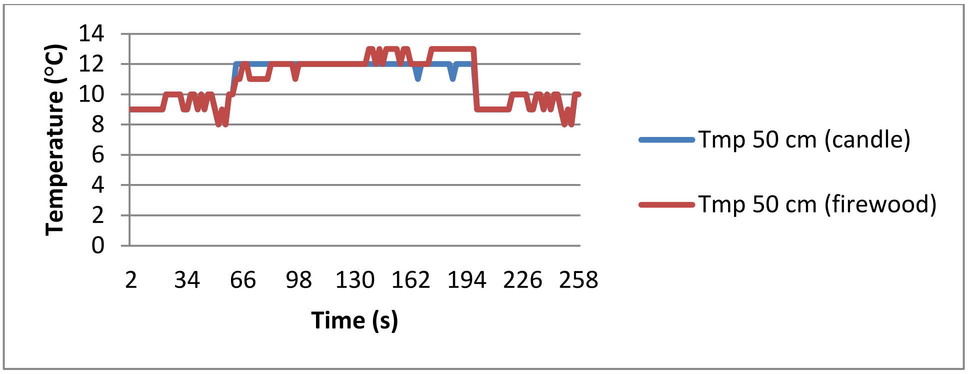


Figure 4.33. Temperature detection comparison between candle and firewood flame

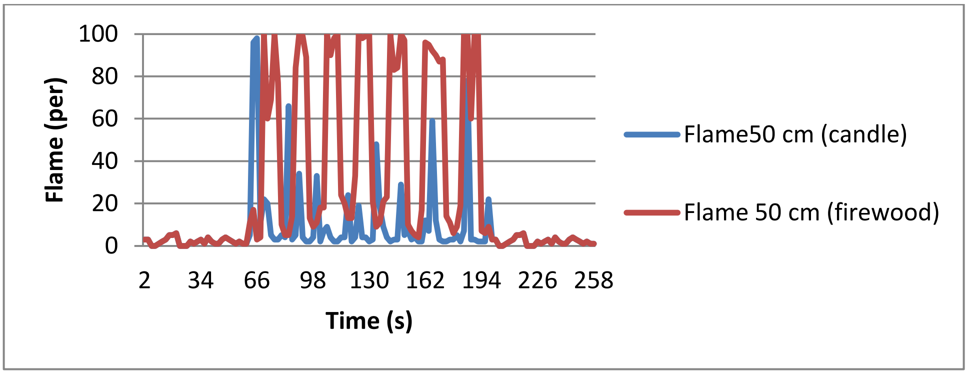


Figure 4.34. Flame detection comparison between candle and firewood flame

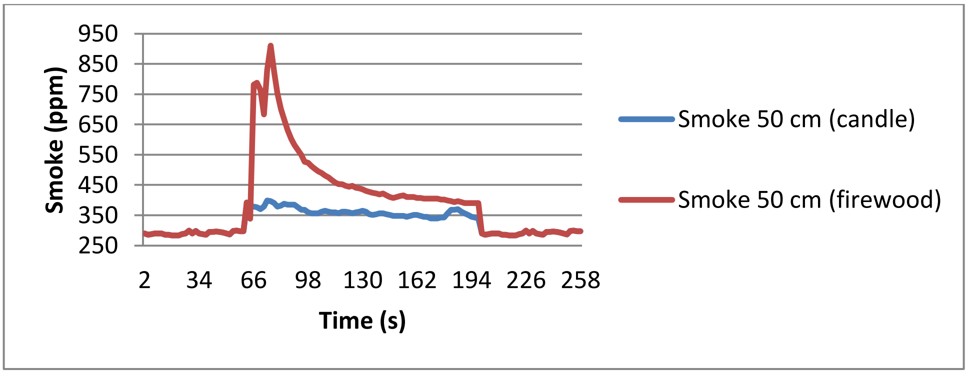


Figure 4.35. Smoke detection comparison between candle and firewood flame

It was understood from the Figures 4.27 and 4.30 that the distance has very little effect on the variation of measured temperature values. The temperature of the non-fire environment was about 10˚C and it raised up to 12 ˚C with candle flame. This maximum value was measured from all of the distances. The maximum value of the firewood flame 13 ˚C was measured from 50 cm and 75 cm distances.

As the flame sensor was operated with a servo motor and it had 210˚ (180˚ comes from servo motor and 30˚ is the feature of the sensor) scanning angle, fluctuations were seen in the flame detection values. The distance and flame size effects could be seen in Figures 4.28 and 4.31. Sensor could sense the firewood flame better than the candle flame as it had larger size and higher detection values were seen more frequent in the distance of 50 cm; it means that the detection quality increases as the distance decreases.

The measured smoke value of the non-fire environment was about 290 ppm and maximum values were 400 ppm for candle, 900 ppm for firewood. It was understood from the Figures 4.29 and 4.32 that the distance and fire source affected the detection quality.

As the result of these tests, we determined that distance and type of fire source affected the quality of fire detection. Although system had ability to detect fire in the range between 10 cm and 100 cm, we specified the minimum detection distance as 50 cm, because of the safety reason and prevent the robot damage.

Sample screenshot from the test is shown in Figure 4.36.

Figure 4.36. Fire detection based on distance test

4.3.2. Detection Test on Patrolling

In this test; the reaction of the fire detection unit was investigated while robot was tracking its path and fire source was within detection range. Candle and firewood were used as fire sources. The aim of this test was to determine the fire detection capability of the system when it executes patrolling task with 0.5 m/s speed. While robot was moving, the efficiency of the sensors was experimented.

Experiments were carried out under controlled outdoor conditions and at evening hours to prevent interference between flame and sun lights. A test environment which has the same dimensions with path tracking test was established. At first step of the test; the intense of IR light, smoke and temperature values of the non-fire environment were measured for 60 seconds via sensors and saved to netbook. At the next step, candle flame was located at 50 cm distance from path tracking route. The variation of IR light, smoke and temperature values were measured and saved for 140 seconds while robot is tracking its path. The same process was applied for firewood flame. Relevant obtained results are summarized in Figures 4.37 to 4.42.

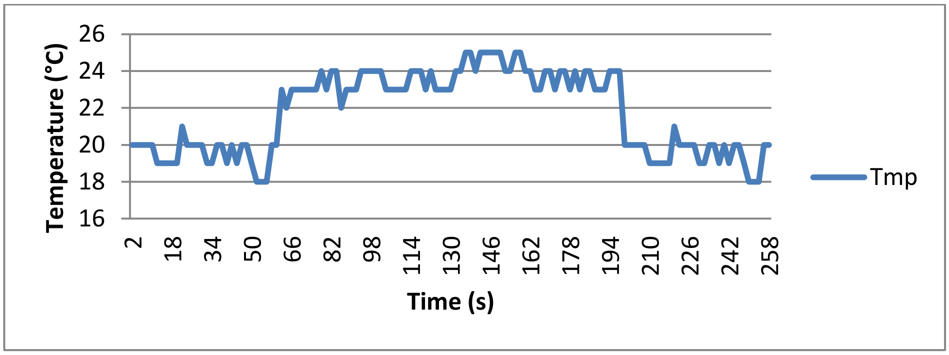


Figure 4.37. Temperature sensor values for candle flame-mobile mode

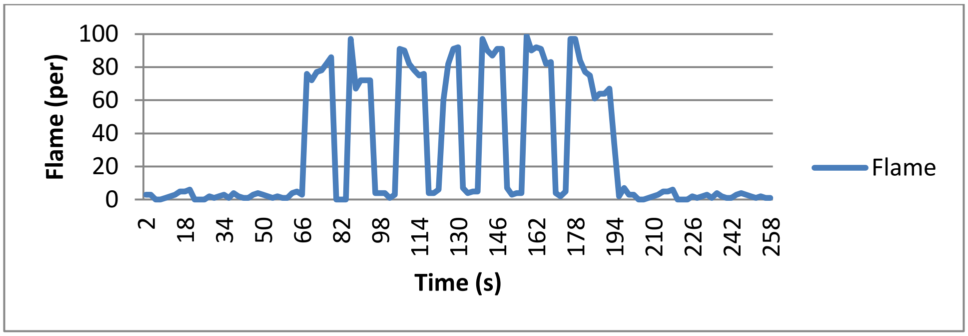


Figure 4.38. Flame sensor values for candle flame-mobile mode

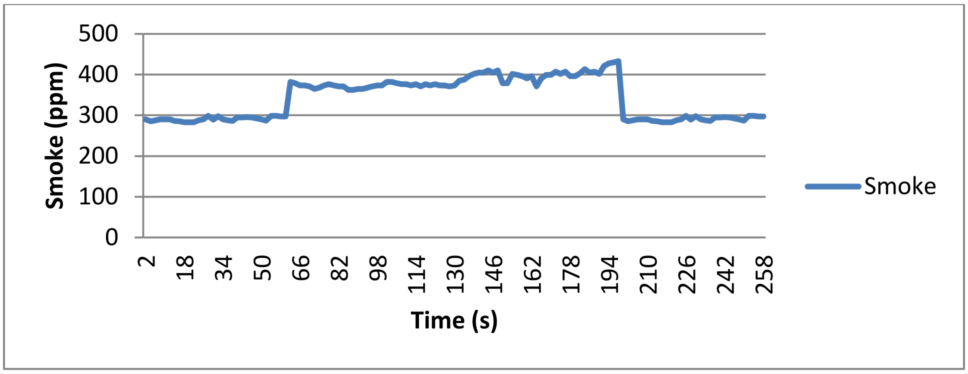


Figure 4.39. Smoke sensor values for candle flame-mobile mode

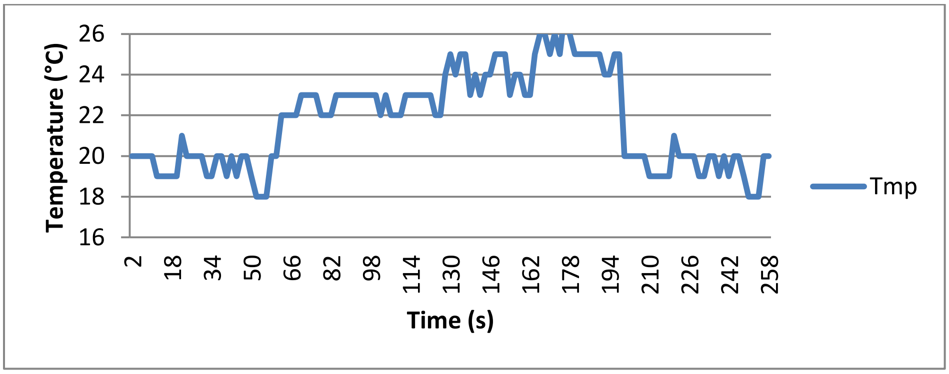


Figure 4.40. Temperature sensor values for firewood flame-mobile mode

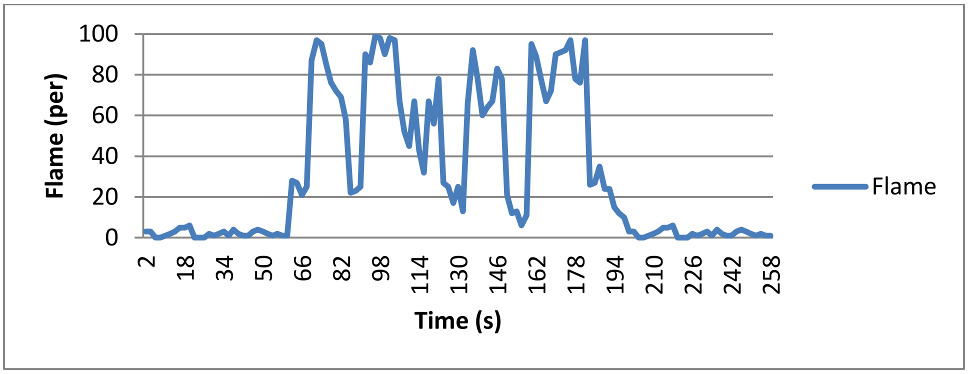


Figure 4.41. Flame sensor values for firewood flame-mobile mode

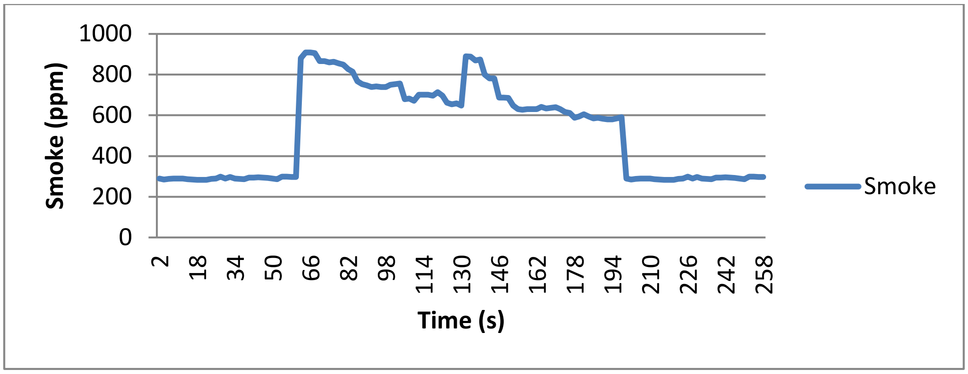


Figure 4.42. Smoke sensor values for firewood flame-mobile mode

It was observed from the tests that smoke and flame sensors could detect the candle and firewood flames when the sources were in the detection range. While the non-fire smoke values were about 300 ppm; sensor values reached 400 ppm level for candle flame and 900 ppm level for firewood flame. Flame sensor values raised from per 10 to 100. It means that sensors could sense the smoke and flame in the range of patrolling area. As a result, fire detection robot could satisfy the fire detection requirements while it was executing the patrolling task. Sample screenshot from the test is shown in Figure 4.43.

Figure 4.43. Mobile fire detection test

4.3.3. Sensor Data Fusion Algorithm Test

To get more reliable fire detection results, three sensors were used in the system and a data sensor data fusion algorithm was developed using the results of fire detection test based on distance.

Sensor data fusion algorithm was tested using the equation 3.19. Three different terms were defined for fire detection. The values of these terms are given in Table 4.2.

Table 4.2. Terms and values of fire detection

Name of the situation	**Value**
No fire	0-0.3
Danger	0.3-0.6
Fire	0.6-1

Three Leds (Light emitted diode) red, yellow and green were connected to fire detection unit circuit and fire detection algorithm equation was programmed to microcontroller. The algorithm was tested five times for five different distances using a lighter. The details of the algorithm are given below;

1. If the calculated fire value is in the range of 0-0.3 the green Led will be turned on (No fire situation).

2. If the calculated fire value is between 0.3-0.6 the yellow Led will be turned on (Danger situation).

3. If the calculated fire value is between 0.6-1 the red Led will be turned on (Fire situation).

The results of the test are given in Table 4.3.

Table 4.3. Results of the fire detection test

Distances	**Performance of the algorithm**				
	Test 1	**Test 2**	**Test 3**	**Test 4**	**Test 5**
10 cm	Red	Red	Red	Red	Red
50 cm	Red	Red	Red	Red	Red
60 cm	Red	Red	Red	Red	Yellow
80 cm	Red	Red	Red	Yellow	Red
100 cm	Red	Red	Red	Red	Green

As seen in the Table 4.3 system could detect the fire with hundred percent performances for 10 cm and 50 cm distances. For the 60 and 80 cm distances, system answered the lighter flame as four time fire and one time danger. One false alarm was shown in the 100 cm distance. Although there was a simulated fire, system answered as no fire. The results indicated that the developed sensor data fusion algorithm could answer the fire source with hundred percent performances in the range of 10-80 cm distances. For the 100 cm distance there was only one wrong answer. It can be said that the data fusion algorithm is able to produce reliable results. Sample screenshot is shown in Figure 4.44.

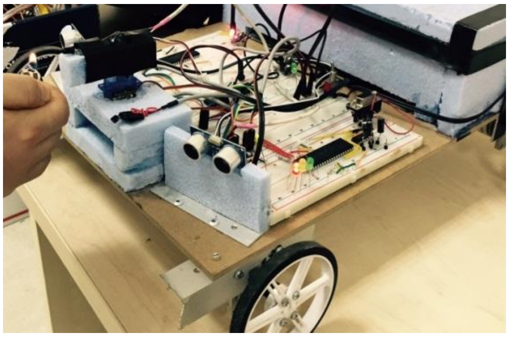


Figure 4.44. Fire detection algorithm test

Some related works are compared with our fire detection robot. These comparisons are given below.

Khoon et al. (2012) developed an autonomous fire detection mobile platform. Comparison between Khoon's and our system is given in Table 4.4.

Table 4.4. Comparison with Khoon's system

	Khoon's system	**Our system**
Path tracking methods	Physical line tracking with LDRs and LEDs	Virtual path tracking with Re-programmable motion plan and control unit
Fire detection sensors	Flame sensor	Flame, temperature and smoke sensors
Environment scanning	System stops and turns around for 360° to scan the environment.	Flame sensor scans the environment for 210° with servo motors while robot is executing patrolling task.
Test method	Only flame sensor's voltage outputs based on distances are tested.	Percent, temperature and ppm outputs of the sensors are tested for both stationary and mobile situations.

Xu et al. (2011) developed a mobile robot remote fire alarm system. Comparison between Xu's and our system is given in Table 4.5.

Table 4.5. Comparison with Xu's system

	Xu's system	**Our system**
Path tracking methods	Physical line tracking with photoelectric sensors	Virtual path tracking with Re-programmable motion plan and control unit
Fire detection sensors	Although flame, temperature and smoke sensors are used in the system, fire is detected only using flame sensor. Temperature and smoke sensors are used for environmental measurements.	A fire estimation model is developed using the measurement values of flame, temperature and smoke sensors.
Obstacle avoidance methods	IR sensors are used to detect obstacle. Robot is programmed to stop when there is an obstacle.	Ultrasound sensors are used for obstacle avoidance. Robot is programmed to search for free path and pass the obstacle.

Roberto et al. (2013) proposed fire detection method using flame, temperature and luminosity sensors through a mobile robot platform. Comparison between Roberto's and our system is given in Table 4.6.

Table 4.6. Comparison with Roberto's system

	Roberto's system	**Our system**
Fire detection sensors	Flame, temperature and luminosity sensors	Flame, temperature and smoke sensors
Fire detection methods	Local data fusion approach and moving average filters is used for fire detection.	Sensor data fusion algorithm and PLS analysis are used to specify weight values of the sensors.
Tested fire sources	Alcohol, paper	Candle, firewood
Test methods	Stationary method	Stationary and mobile methods
Detection distances	100 cm is determined as best detection distance.	50 cm is determined as best detection distance.

5. CONCLUSIONS

This thesis is about to design and produce a fire detection mobile robot which can operate especially in industrial areas and to test its system functions.

Throughout this study; design and manufacturing process are explained, functions of the robot are tested and results are discussed. The manufactured robot is a prototype of the planned advanced firefighting robot. The budget of this thesis is limited and advanced manufacturing machines, sensors and hardware cannot be used, so the robot is produced using the cheapest components that can be found in the market easily. Despite the disadvantages, fire detection robot passed the functions tests successfully. It executed the patrolling task easily, passed the obstacle without any collision or interruption, detected the fire up to 100 cm for stationary mode and up to 50 cm for mobile mode and its sensor data fusion algorithm answered the fire correctly.

6. RECOMMANDATIONS AND FUTURE WORKS

By further developments and improvements, fire detection robot will have more functions and produce more reliable fire detection results. To improve the fire detection robot functions some additions and changes should be done.

Chassis and the main body of the robot can be manufactured using fire proof material. With the fireproof material, robot can get into fire. The smoke, temperature and flame measurements can be carried out closer to fire source so that more reliable detection results can be obtained.

RTK-GPS can be used for motion planning and navigation of the robot. With RTK-GPS receiver and a navigation algorithm the operation system of the robot is improved from semi-autonomous to autonomous. It can be operate at unstructured environments easily.

Robot can be equipped with advanced fire detection equipment. With a minor change; using more flame sensor instead of scanning the environment only one flame sensor and servo motor, more reliable flame detection results can be obtained. To produce an advanced level of firefighting robot which can operate at real fire scenarios; combination of optical flame detectors, thermal camera and image processing are required.

Wireless communication module and camera system can be integrated to robot so that it can send information about the fire site and communicate between the victims and operator.

As future research directions, we intend to develop a prototype autonomous-unmanned equipment that can intervene field, forest and open area fire, and that can reach critical places which cannot be reached by the conventional vehicles so as to make first intervention, and to support rescuing operations during the fire incidents; and to test the equipment for its performance within the virtual and real time environments.

REFERENCES

Anonymous (2005). Fire in Workplace. **MOC Safety Newsletter.**

Anonymous (12.01.2007). Eleventh Occasional Forensic Genealogy Survey [http://www.forensicgenealogy.info/survey_11_results.html], Last accessed in August 2014.

Anonymous (07.03.2008). Litz Wire Products Information Page, Protectowire Linear Heat Detectors [http://hflitzwire.com], Last accessed in July 2014.

Anonymous (10.04.2009). Desu Product Information Guidance, Frequently Asked Questions About Flame Detection [http://www.desusystems.com/library], Last accessed in July 2014.

Anonymous (2010 a). Big Industrial Fires [http://www.iklimnet.com/hotelfires/big_industrial_fires.html], Last accessed in July 2014.

Anonymous (01.02.2010 b). Robot Platform, Classification of Robots [http://www.robotplatform.com/knowledge/Classification_of_Robots/wheel_control_theory.html], Last accessed in September 2014.

Anonymous (12.05.2010 c). Apollo Product Information Guidance, Frequently Asked Questions About Smoke Detection [http://www.apollo-fire.co.uk], Last accessed in July 2014.

Anonymous (07.03.2011 a). Mobile Firefighting Supporting Machine LUF60 Product Page [http://www.hkfsd.gov.hk/images/equipment/fire/e_luf60.html], Last accessed in July 2014.

Anonymous (11.05.2011 b). MAX232 IC. [http://www.engineersgarage.com/electroniccomponents/max232datasheet], Last accessed in December, 2014.

Anonymous (02.10.2011 c). HC-SR04 Ultrasonic Sensor Distance Measuring Module Product Page [http://users.ece.utexas.edu/~valvano/Datasheets/HCSR04b.pdf], Last accessed in October 2014.

Anonymous (09.10.2011 d). Texas Instruments, LM 35 Temperature Sensor Product Page [http://www.ti.com/lit/ds/symlink/lm35.pdf], Last accessed in October 2014

Anonymous (2012, a). International Federation of Robotics, Service Robot Statistics [http://www.ifr.org/servicerobots/statistics/], Last accessed in September 2014.

Anonymous (2012, b). International Federation of Robotics, Industrial Robot Statistics [http://www.ifr.org/industrial-robots/statistics/], Last accessed in September, 2014.

Anonymous (2012, c). Robots and Robotic Devices [https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en], Last accessed in August 2014.

Anonymous (07.03.2012 d). Armtec Technology Product Information Page [http://v2.armtecbrasil.com/], Last accessed in July 2014.

Anonymous (01.04.2012 e). ICCN EXERGY web page, Rechargeable Lithium-ion Batteries [http://www.iccnexergy.com/battery-systems/rechargeable-lithium/], Last accessed in October 2014.

Anonymous (04.07.2012 f). Atmel Company, Atmega 32 Datasheet [http://www.atmel.com/images/doc2503.pdf], Last accessed in December, 2014.

Anonymous (07.01.2013 a). DRB Fatec Product Information Page [http://www.drbfatec.com/], Last accessed in July 2014.

Anonymous (05.02.2013 b). Thermite Robot Product Information Page [http://www.howeandhowe.com/], Last accessed in July 2014.

Anonymous (06.02.2013 c). MVF 5 Product Information Page [http://dok-ing.hr/products/firefighting/mvf_5], Last accessed in July 2014.

Anonymous (06.06.2013 d). Shadow Robot Company Product Information Guidance, Dexterous Hand [http://www.shadowrobot.com/products/dexterous-hand/], Last accessed in September 2014.

Anonymous (07.09.2013 e) . LM 7805 Voltage Regulator IC [http://www.engineersgarage.com/electronic-components/7805-voltage-regulator-ic], Last accessed in December, 2014.

Anonymous (01.10.2013 f). High Capacity Unmanned Firefighting Vehicle, Ryland Research Limited Product Information Page [http://www.rylandresearch.co.uk/remotely-operated-vehicles/firemote-4800], Last accessed in July 2014.

Anonymous (05.10.2013 g). Pololu Carbon Monoxide & Flammable Gas Sensor MQ-9 Product Page [http://www.pololu.com/product/1483], Last accessed in October 2014.

Anonymous (12. 10.2013 h). Pololu MQ-9 Carrier Board Product Page [http://www.pololu.com/product/1479], Last accessed in October 2014.

Anonymous (07.10.2013 i). Arduino Flame Sensor Product Page [http://www.fut-electronics.com/wp-content/plugins/fe_downloads/Uploads/Flame-sensor arduino.pdf], Last accessed in October 2014.

Anonymous (10.11. 2013 j). How to Use Mosfet? [http://blog.oscarliang.net/how-to-use-mosfet-beginner-tutorial/], Last accessed in December, 2014.

Arkin, R.C. (Ed.) 1998. Whence Behaviour? In: Behaviour Based on Robotics, Massachusetts Institute of Technology pp.1-28, United States of America.

Behnke, S. 2008. Humanoid Robots From Fiction to Reality. **CiteSeer,** pp.5-9.

Bekem, I., Cavus, M., Demirel, F. 2011. Türkiye Ölçeğinde Yangın İstatistikleri Üzerine Bir Çalışma. **Tüyak Yangın ve Güvenlik Sempozyumu ve Sergisi**.

Breazeal, C., Berlin, M., Brooks, A., Gray, J., Thomaz, A.L. 2006. Using perspective taking to learn from ambiguous demonstrations. **Robotics and Autonomous Systems**, 54: 385-393.

Brooks, R.A., Flynn, A.M. 1989. Fast, cheap and out of control: A robot invasion of the solar system. **Journal of the British Interplanetary Society,** 42: 478-485.

Brushlinsky, N.N., Sokolo, S.V., Wagner I. P., Hall, J.R. 2006.World Fire Statistics. **Centre of Fire Statistic of CTIF,** Report 10.

Ceccarelli, M., Carbone, G. 2005. Legged Robotic Systems. In: Cutting Edge Robotics (Vedran, K., Aleksandar, L., Munir, M.), InTech, pp. 553-576, Germany.

Chang, P.H., Kang, Y.H., Cho, G.R., Kim, J.H., Jin, M., Lee, J. 2006. Control Architecture Design for a Fire Searching Robot using Task Oriented Design Methodology. **SICE-ICASE International Joint Conference, IEEE,** pp.3126 - 3131, Busan.

Chen, X.Q., Chen, Y.Q., Chase, J.G. (Eds.) 2009. Mobiles Robots – Past Present and Future. In: Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions, InTech, pp.1-32.

Dan, K. 2012. How Does an SPDT Relay Work [http://www.ehow.com/how-does_4911787_spdt-relay-work.html], Last accessed in October 2014.

De Santos, P.G., Garcia, E., Estremera, J. (Ed.) 2006. Dynamic Stability Criteria. In: Quadrupedal Locomotion, Springer, pp.37-42, Germany.

Durán, B., Thill, S. 2012. Rob's Robot: Current and Future Challenges for Humanoid Robots. In: The Future of Humanoid Robots - Research and Applications (Riadh, Z.), InTech, pp. 280-300.

Garcia, E., Jimenez, M., De Santos, P., Armada, M. 2007. The Evolution of Robotics Research. **Robotics & Automation Magazine, IEEE,** 14: 90-103.

Geppert, L. 2014. The Robot That Could. **Spectrum, IEEE,** 41:34-37

Gupta, P., Tirth, V., Srivastava, R.K. 2006. Futuristic Humanoid Robots: An Overview. **International Conference on Industrial and Information Systems, IEEE,** pp. 247 - 254, Peradeniya.

Hahnel, D., Triebel, R., Burgard, W., Thrun, S. 2003. Map Building with Mobile Robots in Dynamic Environments. **International Conference on Robotics and Automation, IEEE**, pp. 1557 - 1563.

Haukur, I., Heimo, T., Anders, L. 2010. Industrial fires An Overview, **Brandforsk Project.**

Hirose, S., Kato, K. 2000. Study on Quadruped Walking Robot in Tokyo Institute of Technology-Past, Present and Future. **International Conference on Robotics and Automation,** IEEE, pp.414-149, San Francisco.

Hobbelen, D.G.E., Wisse, M. 2007. Limit Cycle Walking. In: Humanoid Robots, Human-like Machines (Matthias, H.), InTech, pp.278-294, China.

Hrbáček, J., Ripel, T., Krejsa, J.2010. Ackermann mobile robot chassis with independent rear wheel drives. **14th International Power Electronics and Motion Control Conference, IEEE,** pp.46-51, Ohrid.

Kar, D.C. 2003. Design of Statically Stable Walking Robot: A Review. **Journal of Robotic Systems,** 11:671-686.

Karl, M.(Ed.) 2010. Industrial Robotics. In Robotics for Electronic Manufacturing, Cambridge University Press, pp.1-11, New York.

Khoon, T.N., Sebastian, P., Saman, A.B.S.2012. Autonomous Fire Fighting Mobile Platform. **Procedia Engineering,** 41:1145-1153.

Kim, Y.D., Kim, Y.G., Lee, S-H., Kang, J-H., An, J. 2009. Portable Fire Evacuation Guide Robot System. **International Conference on Intelligent Robots and Systems**, pp. 2789 - 2794, St. Louis.

Kumar, P.S, Ratheesh, B.R., Gobinath, B., Kumaran, K.M., Venkataraman, S. 2007. Gesture Controlled Fire Extinguisher Robot with Audio and Video Capture. **IOSR Journal of Electronics and Communication Engineering,** pp.101-105.

Ligo, G. 2012. ULN 2003 High Voltage and High Current Darlington Transistor Array [http://electrosome.com/uln2003-high-voltage-current-driver/], Last accessed in December, 2014.

Lim, H-O., Takanishi, A. (Eds.) 2006. Mechanism and Control of Anthropomorphic Biped Robots. In: Mobile robots, moving intelligence, InTech, pp. 308-324, Austria.

Lund, H.H., Miglino O. 1996. From Simulated to Real Robots. **International Conference on Evolutionary Computation, IEEE,** pp. 362 - 365, Nagoya.

Luo, R.C., Su, K.L. 2007. Autonomous Fire-Detection System Using Adaptive Sensory Fusion for Intelligent Security Robot. **Ieee/Asme Transactions on Mechatronics,** 12:274-281.

Margaret, R. 2005. What is a Diode? [http://whatis.techtarget.com/definition/diode], Last accessed in October 2014.

Marlene, F. (02.03.2014). Agriculture and Rural Development, Track Versus Wheel,[http://www1.agric.gov.ab.ca/$department/deptdocs.nsf/all/eng8171] Last accessed in September 2014.

Marshall, B., Charles, W.B.2010. How Capacitor Work? [http://electronics.howstuffworks.com/capacitor.html], Last accessed in December, 2014.

Martinson, E., Lawson, W., Blisard, S., Harrison, A., Trafton, G. 2012. Fighting Fires with Human Robot Teams. **International Conference on Intelligent Robots and Systems, IEEE,** pp.2682 - 2683, Vilamoura.

Matsui, D., Minato, T., MacDorman, K.F., Ishiguro, H. 2005. Generating Natural Motion in an Android by Mapping Human Motion. **International Conference on Intelligent Robots and Systems, IEEE,** pp. 3301-3308.

Minato, T., Yoshikawa, Y., Noda, T., Ikemoto, S., Ishiguro, H., Asada, M.2007. CB2 A Child Robot with Biomimetic Body for Cognitive Developmental

Robotics. **International Conference on Humanoid Robots, IEEE,** pp. 557-562, Pittsburgh.

Mishra, K. B., Wehrstedt, K.D., Krebs, H. 2013. Lessons learned from recent fuel storage fires. **Fuel Processing Technology,** 107: 166 – 172.

Narukawa, T., Yokoyama, K., Takahashi, M., Yoshida, K. 2010. An experimental study of three dimensional passive dynamic walking with flat feet and ankle springs. In: Cutting Edge Robotics (Vedran, K.), InTech, pp.132-144, China.

Naunheimer, H., Bertsche, B., Ryborz, J., Novak, W. (Eds.) 2011. Mediating The Power Flow. In: Automotive Transmissions, Springer, pp. 58-75, New York.

Nehmzow, U. (Ed.) 2003. A Practical Introduction. In: Mobile Robotics, Springer pp. 25-45, New York.

Nishio, S., Ishiguro, H., Hagita, N. 2007. Geminoid: Teleoperated Android of an Existing Person. In: Humanoid Robots: New Developments (Armando C.P. F.), InTech, pp.344-352, China.

Nonami, K., Barai, R.K., Irawan, A., Daud, M.R. (Eds.) 2014. Historical and Modern Perspective of Walking Robots. In: Hydraulically Actuated Hexapod Robots, Springer, pp.19-40, New York.

Ozguner, F., Tsai, S.J. 1985. Design and Implementation of a Binocular-Vision System for Locating Footholds of a Multi-Legged Walking Robot. **Neural Networks, IEEE Transactions,** 32:26-31.

Park, W., Kim, J.Y., Lee, J., Oh, J.H. 2005. Mechanical Design of Humanoid Robot Platform KHR-3 (KAIST Humanoid Robot-3:HUBO). **International Conference on Humanoid Robots, IEEE,** pp.321-326, Tsukuba.

Patnaik, S. (Ed.) 2007. Cybernatic View of Robot Cognition and Perception. In: Robot Cognition and Navigation, Springer, pp 10-20, New York.

Porteous, R. 2011. Working Principles of Heat and Smoke Detectors, [http://firewize.com], Last accessed in July 2014.

Reeve, R., Hallam J. 2005. Analysis of Neural Models for Walking Control. **Neural Networks, IEEE Transactions,** 16:733-742.

Richardson, K. 2008. The fact and fiction of robots. **The Thinking Machine,** University of Cambridge.

Roberto, G.F., Branco, K.C., Machado, J.M., Pinto, A.R. 2013. Local Data Fusion Algorithm for Fire Detection through Mobile Robot. **Test Workshop (LATW), 14th Latin American**, pp.1-6, Cordoba.

Rosheim, M.E. (Ed.) 1994. Robots past. In: Robot Evolution: The Development of Anthrobotics, Wiley & Sons, Inc., pp.1-37, New York, USA.

Sciavicco, L., Siciliano, B. (Eds.) 2005. Robotics. In: Modelling and Control of Manipulators, McGraw Hill, pp.1-16, London.

Shamah, B. 1999. Experimental Comparison of Skid Steering vs. Explicit Steering for a Wheeled Mobile Robot, **CiteSeer.**

Silva, M.F., Machado, J.A.T. 2007. A Historical Perspective of Legged Robots. **Journal of Vibration and Control,** 20: 1378-1393.

Tajti, F., Szayer, G., Kovacs, B., Daniel, B., Korondi P. 2013. CRM TC covering paper - Robotics Trends. **Industrial Electronics Society, IECON 39th Annual Conference of the IEEE,** pp.48-53, Vienna.

Tan, C.F., Liew, S.M., Alkahari, M.R., Ranjit, S.S.S., Said, M.R., Chen W., Rauterberg, G.W.M., Sivarao D.S. 2013. Fire Fighting Mobile Robot: State of the Art and Recent Development. **Australian Journal of Basic and Applied Sciences,** 10: 220-230.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P. 2006. Stanley: The Robot that Won the DARPA Grand Challenge. **Journal of Field Robotics,** pp. 661-692.

Tongying, G., Ma, B., Li, J., He, H. 2010. Research on Flame Image Edge Detection Algorithm Based on Mobile Robot Platform. **International Conference on Artificial Intelligence and Computational Intelligence, IEEE,** pp. 78 - 81, Sanya.

Valgren, C. 2007. Incremental Spectral Clustering and Its Application To Topological Mapping. **International Conference on Robotics and Automation, IEEE,** pp. 4283-4288, Roma.

Vukobratovic, M., BOROVAC B. 2004. Zero Moment Point Thirty five years of its life. **International Journal of Humanoid Robotics,** 1:157-173.

Wallén, J. 2008. Estimation-based iterative learning control. Department of Electrical EngineeringLinköping University,Ph.D.Thesis, Linköping.

Wayne, D.M. 2006. Fire Alarm System Research-Where it's been and where it's going. **Fire Suppression & Detection Research Application Symposium.**

Xu, H., Chen, H., Cai, C., Guo, X., Fang, J., Sun, Z. 2011. Design and Implementation of Mobile Robot Remote Fire Alarm System. **International Conference on Intelligence Science and Information Engineering,** pp. 32 - 36, Wuhan.

Zhang, F., Li, M., Liang, P., Zhang, K., Chou, Y. 2012. The Design And Implementation Of Intelligent Inspection Robot System Using Combined Flame Sensors And Infrared Sensors. **International Symposium on Instrumentation & Measurment, Sensor Network and Automation (IMSNA), IEEE,** pp. 297-301, Sanya.

Zhang, W., Dai C. 2009. Development of A New Remote Controlled Emergency-handling and Fire-fighting Robot. **World Congress on Computer Science and Information Engineering, IEEE,** pp. 239 - 243, Los Angeles.

Zielinska, T. 2004. Development of a walking machine: mechanical design and control problems. **Mechatronics,** 12:737-754

APPENDICES

Appendix-1 (Microcontroller Codes)

Path Tracking With 0.2 m/s Forward Speed

```
#include <mega32a.h>

#include <delay.h>

void moveForward(int gear) {

  int time=0;

  int operatingTime=0;

  int waitingTime=0;

   if (gear==0) {

      PORTC.7=0; //Right Motor Off

      PORTC.6=0; //Left Motor Off

      delay_ms(100);

   } else if (gear==5) {

      PORTC.7=1; //Right Motor On

      PORTC.6=1; //Left Motor On

      delay_ms(100);

   }

   else {

   time=gear*20;

   operatingTime=time;

   waitingTime=100-time;

      //void DCMotorControl
```

```
    // Place your code here
    PORTC.7=1; //Right Motor On
    PORTC.6=1; //Left Motor On
delay_ms(operatingTime);
    PORTC.7=0; //Right Motor Off
    PORTC.6=0; //Left Motor Off
    delay_ms(waitingTime);
    }
}
void moveRight(int gear) {
     PORTC.7=1; //Right Motor On
   PORTC.6=0; //Left Motor Off
   delay_ms(20*gear);
}
void moveLeft(int gear) {
   PORTC.7=0; //Right Motor Off
   PORTC.6=1; //Left Motor On
   delay_ms(20*gear);
}
//UltraSound Right
    int ultraSoundRight() {
       unsigned short int tempValue , receivedValue;
       PORTC.0 = 1; // Trigger for 15 uSec
       delay_us(15);
```

```
    PORTC.0 = 0;
        delay_us(100);
    tempValue = 10;  // for waiting rising edge of echo
    while((PINC.1 == 0)&&(tempValue <=2000))
    {
      delay_us(1);
      tempValue = tempValue+1;
    }
    receivedValue = 10;
    while((PINC.1 == 1)&&(receivedValue <=60000))
    {
      delay_us(1);
      receivedValue = receivedValue+1;
    }
    receivedValue=receivedValue/22;
    return receivedValue;
   }
//UltraSound Left
  int ultraSoundLeft() {
    unsigned short int tempValue , receivedValue;
    PORTC.2 = 1; // Trigger for 10 uSec
    delay_us(15);
    PORTC.2 = 0;
    delay_us(100);
```

```
        tempValue = 10;  // for waiting rising edge of echo
        while((PINC.3 == 0)&&(tempValue <=2000))
        {
          delay_us(1);
          tempValue = tempValue+1;
        }
        receivedValue = 10;
        while((PINC.3 == 1)&&(receivedValue <=60000))
        {
          delay_us(1);
          receivedValue = receivedValue+1;
        }
        receivedValue=receivedValue/22;
        return receivedValue;
       }
void turnLeft(){
     moveLeft(135);   // turn 90 left
     moveForward(0);
     delay_ms(3000);
}
void turnRight(){
     moveRight(135);   // turn 90 degree right
     moveForward(0);
     delay_ms(3000);
```

```
}

void main(void)

{

// Local variables

unsigned short int receivedValueLeftUs, receivedValueRightUs;

int gear=0;

int i=0;

// Input/Output Ports initialization

// Port A initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) |
(0<<DDA2) | (0<<DDA1) | (0<<DDA0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) |
(0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization

// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out
Bit0=Out

DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) |
(1<<DDB2) | (1<<DDB1) | (1<<DDB0);

// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0

PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) |
(0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

// Port C initialization

// Function: Bit7=Out Bit6=Out Bit5=In Bit4=In Bit3=In Bit2=Out Bit1=In
Bit0=Out
```

DDRC=(1<<DDC7) | (1<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (1<<DDC2) | (0<<DDC1) | (1<<DDC0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=0

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer1 Stopped

// Mode: Normal top=0xFFFF

// OC1A output: Disconnected

```
// OC1B output: Disconnected

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);

TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: Timer2 Stopped

// Mode: Normal top=0xFF

// OC2 output: Disconnected

ASSR=0<<AS2;
```

```
TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);

TCNT2=0x00;

OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);

MCUCSR=(0<<ISC2);

// USART initialization

// USART disabled

UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

// Analog Comparator initialization

// Analog Comparator: Off

// The Analog Comparator's positive input is

// connected to the AIN0 pin

// The Analog Comparator's negative input is

// connected to the AIN1 pin

ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);

SFIOR=(0<<ACME);
```

```
// ADC initialization

// ADC disabled

ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) |
(0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization

// SPI disabled

SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) |
(0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization

// TWI disabled

TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

while (1)

    {

  gear = 3;  //Forward Speed

    for (i=0;i<100;i++) {   //move forward 8.5 sec

    moveForward(gear);

    }

    moveForward(0);  // stop 2 sec

    delay_ms(2000)

    turnLeft();

    for (i=0;i<60;i++) {   //if its 30 it means 3 sec

    moveForward(gear);

    }

    moveForward(0); // stop 2 sec

    delay_ms(2000)
```

```
      turnLeft();
      for (i=0;i<100;i++) {   //move forward 8.5 sec
      moveForward(gear);
      }
      moveForward(0);   //stop 2 sec
      delay_ms(2000)
      turnLeft();
      for (i=0;i<60;i++) {   //if its 30 it means 3 sec
      moveForward(gear);
      }
      moveForward(0); // stop 2 sec
      delay_ms(2000);
      turnLeft();
   }
}
```

Path Tracking With 0.3 m/s Forward Speed

```
#include <mega32a.h>
#include <delay.h>
void moveForward(int gear) {
    int time=0;
    int operatingTime=0;
    int waitingTime=0;
      if (gear==0) {
          PORTC.7=0; //Right Motor Off
```

```
        PORTC.6=0; //Left Motor Off
        delay_ms(100);
    } else if (gear==5) {
        PORTC.7=1; //Right Motor On
        PORTC.6=1; //Left Motor On
        delay_ms(100);
    }
    else {
    time=gear*20;
    operatingTime=time;
    waitingTime=100-time;
    //void DCMotorControl
    // Place your code here
    PORTC.7=1; //Right Motor On
    PORTC.6=1; //Left Motor On
    delay_ms(operatingTime);
    PORTC.7=0; //Right Motor Off
    PORTC.6=0; //Left Motor Off
    delay_ms(waitingTime);
    }
}
void moveRight(int gear) {
   PORTC.7=1; //Right Motor On
   PORTC.6=0; //Left Motor Off
```

```
    delay_ms(20*gear);
}
void moveLeft(int gear) {
    PORTC.7=0; //Right Motor Off
    PORTC.6=1; //Left Motor On
    delay_ms(20*gear);
}
//UltraSound Right
      int ultraSoundRight() {
          unsigned short int tempValue , receivedValue;
          PORTC.0 = 1; // Trigger for 10 uSec
          delay_us(15);
          PORTC.0 = 0;
          delay_us(100);
          tempValue = 10;  // for waiting rising edge of echo
          while((PINC.1 == 0)&&(tempValue <=2000))
          {
            delay_us(1);
            tempValue = tempValue+1;
          }
          receivedValue = 10;
          while((PINC.1 == 1)&&(receivedValue <=60000))
          {
            delay_us(1);
```

```
      receivedValue = receivedValue+1;
    }
    receivedValue=receivedValue/22;
    return receivedValue;
  }
//UltraSound Left
  int ultraSoundLeft() {
    unsigned short int tempValue , receivedValue;
    PORTC.2 = 1; // Trigger for 10 uSec
    delay_us(15);
    PORTC.2 = 0;
    delay_us(100);
    tempValue = 10;  // for waiting rising edge of echo
    while((PINC.3 == 0)&&(tempValue <=2000))
    {
      delay_us(1);
      tempValue = tempValue+1;
    }
    receivedValue = 10;
    while((PINC.3 == 1)&&(receivedValue <=60000))
    {
      delay_us(1);
      receivedValue = receivedValue+1;
    }
```

```
    receivedValue=receivedValue/22;
    return receivedValue;
   }
void turnLeft(){
   moveLeft(135);   // turn 90 degree left
   moveForward(0);
   delay_ms(3000);
}
void turnRight(){
   moveRight(135);   // turn 90 degree right
   moveForward(0);
   delay_ms(3000);
}
void main(void)
{
// Local variables
unsigned short int receivedValueLeftUs, receivedValueRightUs;
int gear=0;
int i=0;
// Input/Output Ports initialization
// Port A initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) |
(0<<DDA2) | (0<<DDA1) | (0<<DDA0);
```

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization

// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out

DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);

// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0

PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

// Port C initialization

// Function: Bit7=Out Bit6=Out Bit5=In Bit4=In Bit3=In Bit2=Out Bit1=In Bit0=Out

DDRC=(1<<DDC7) | (1<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (1<<DDC2) | (0<<DDC1) | (1<<DDC0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=0

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

// Timer/Counter 0 initialization

// Clock source: System Clock

```
// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) |
(0<<CS02) | (0<<CS01) | (0<<CS00);

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer1 Stopped

// Mode: Normal top=0xFFFF

// OC1A output: Disconnected

// OC1B output: Disconnected

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) |
(0<<WGM11) | (0<<WGM10);

TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) |
(0<<CS12) | (0<<CS11) | (0<<CS10);

TCNT1H=0x00;

TCNT1L=0x00;
```

```
ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: Timer2 Stopped

// Mode: Normal top=0xFF

// OC2 output: Disconnected

ASSR=0<<AS2;

TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);

TCNT2=0x00;

OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);

MCUCSR=(0<<ISC2);
```

```
// USART initialization

// USART disabled

UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) |
(0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

// Analog Comparator initialization

// Analog Comparator: Off

// The Analog Comparator's positive input is

// connected to the AIN0 pin

// The Analog Comparator's negative input is

// connected to the AIN1 pin

ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) |
(0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);

SFIOR=(0<<ACME);

// ADC initialization

// ADC disabled

ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) |
(0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization

// SPI disabled

SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) |
(0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization

// TWI disabled

TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

while (1)

    {
```

```
gear = 4;  //Forward Speed 4
for (i=0;i<100;i++) {   //move forward 8.5 sec
moveForward(gear);
}
moveForward(0);  // stop 2 sec
delay_ms(2000);
turnLeft();
for (i=0;i<60;i++) {   //if its 30 it means 3 sec
moveForward(gear);
}
moveForward(0); // stop 2 sec
delay_ms(2000);
turnLeft();
for (i=0;i<100;i++) {   //move forward 8.5 sec
moveForward(gear);
}
moveForward(0);   //stop 2 sec
delay_ms(2000);
turnLeft();
for (i=0;i<60;i++) {   //if its 30 it means 3 sec
moveForward(gear);
}
moveForward(0); // stop 2 sec
delay_ms(2000);
```

```
    turnLeft();
  }
}
```

Path Tracking With 0.5 m/s Forward Speed

```
#include <mega32a.h>
#include <delay.h>
void moveForward(int gear) {
   int time=0;
   int operatingTime=0;
   int waitingTime=0;
    if (gear==0) {
        PORTC.7=0; //Right Motor Off
        PORTC.6=0; //Left Motor Off
        delay_ms(100);
    } else if (gear==5) {
        PORTC.7=1; //Right Motor On
        PORTC.6=1; //Left Motor On
        delay_ms(100);
    }
    else {
    time=gear*20;
    operatingTime=time;
    waitingTime=100-time;
    //void DCMotorControl
```

```
    // Place your code here
    PORTC.7=1; //Right Motor On
    PORTC.6=1; //Left Motor On
    delay_ms(operatingTime);
    PORTC.7=0; //Right Motor Off
    PORTC.6=0; //Left Motor Off
    delay_ms(waitingTime);
    }
}
void moveRight(int gear) {
  PORTC.7=1; //Right Motor On
  PORTC.6=0; //Left Motor Off
  delay_ms(20*gear);
}
void moveLeft(int gear) {
  PORTC.7=0; //Right Motor Off
  PORTC.6=1; //Left Motor On
  delay_ms(20*gear);
}
//UltraSound Right
    int ultraSoundRight() {
      unsigned short int tempValue , receivedValue;
      PORTC.0 = 1; // Trigger for 10 uSec
      delay_us(15);
```

```
    PORTC.0 = 0;
    delay_us(100);
    tempValue = 10;  // for waiting rising edge of echo
    while((PINC.1 == 0)&&(tempValue <=2000))
    {
      delay_us(1);
      tempValue = tempValue+1;
    }
    receivedValue = 10;
    while((PINC.1 == 1)&&(receivedValue <=60000))
    {
      delay_us(1);
      receivedValue = receivedValue+1;
    }
    receivedValue=receivedValue/22;
    return receivedValue;
   }
//UltraSound Left
  int ultraSoundLeft() {
    unsigned short int tempValue , receivedValue;
    PORTC.2 = 1; // Trigger for 10 uSec
    delay_us(15);
    PORTC.2 = 0;
    delay_us(100);
```

```
    tempValue = 10;  // for waiting rising edge of echo
    while((PINC.3 == 0)&&(tempValue <=2000))
    {
      delay_us(1);
      tempValue = tempValue+1;
    }
    receivedValue = 10;
    while((PINC.3 == 1)&&(receivedValue <=60000))
    {
      delay_us(1);
      receivedValue = receivedValue+1;
    }
    receivedValue=receivedValue/22;
    return receivedValue;
   }
void turnLeft(){
   moveLeft(135);   // turn 90 degree left
   moveForward(0);
   delay_ms(3000);
}
void turnRight(){
   moveRight(135);   // turn 90 degree right
   moveForward(0);
   delay_ms(3000);
```

```
}

void main(void)

{

// Local variables

unsigned short int receivedValueLeftUs, receivedValueRightUs;

int gear=0;

int i=0;

// Input/Output Ports initialization

// Port A initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) |
(0<<DDA2) | (0<<DDA1) | (0<<DDA0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) |
(0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization

// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out
Bit0=Out

DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) |
(1<<DDB2) | (1<<DDB1) | (1<<DDB0);

// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0

PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) |
(0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

// Port C initialization

// Function: Bit7=Out Bit6=Out Bit5=In Bit4=In Bit3=In Bit2=Out Bit1=In
Bit0=Out
```

```
DDRC=(1<<DDC7) | (1<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (1<<DDC2) | (0<<DDC1) | (1<<DDC0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=0

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer1 Stopped

// Mode: Normal top=0xFFFF

// OC1A output: Disconnected
```

```
// OC1B output: Disconnected

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);

TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: Timer2 Stopped

// Mode: Normal top=0xFF

// OC2 output: Disconnected

ASSR=0<<AS2;
```

```
TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);

TCNT2=0x00;

OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);

MCUCSR=(0<<ISC2);

// USART initialization

// USART disabled

UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

// Analog Comparator initialization

// Analog Comparator: Off

// The Analog Comparator's positive input is

// connected to the AIN0 pin

// The Analog Comparator's negative input is

// connected to the AIN1 pin

ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);

SFIOR=(0<<ACME);
```

```
// ADC initialization

// ADC disabled

ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) |
(0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization

// SPI disabled

SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) |
(0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization

// TWI disabled

TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

while (1)

    {

    gear = 5;  //Forward Speed 5

    for (i=0;i<100;i++) {   //move forward 8.5 sec

    moveForward(gear);

    }

    moveForward(0);  // stop 2 sec

    delay_ms(2000);

    turnLeft();

    for (i=0;i<60;i++) {   //if its 30 it means 3 sec

    moveForward(gear);

    }

    moveForward(0); // stop 2 sec

    delay_ms(2000);
```

```
      turnLeft();
      for (i=0;i<100;i++) {   //move forward 8.5 sec
      moveForward(gear);
      }
      moveForward(0);   //stop 2 sec
      delay_ms(2000);
      turnLeft();
      for (i=0;i<60;i++) {   //if its 30 it means 3 sec
      moveForward(gear);
      }
      moveForward(0); // stop 2 sec
      delay_ms(2000);
      turnLeft();
  }
}
```

Obstacle Avoidance Algorithm

```
#include <mega32a.h>
#include <delay.h>
void moveForward(int gear) {
   int time=0;
   int operatingTime=0;
   int waitingTime=0;
     if (gear==0) {
        PORTC.7=0; //Right Motor Off
```

```
        PORTC.6=0; //Left Motor Off
    } else if (gear==5) {
        PORTC.7=1; //Right Motor On
        PORTC.6=1; //Left Motor On
    }
    else {
    time=gear*20;
    operatingTime=time;
    waitingTime=100-time;
    //void DCMotorControl
    // Place your code here
    PORTC.7=1; //Right Motor On
    PORTC.6=1; //Left Motor On
    delay_ms(operatingTime);
    PORTC.7=0; //Right Motor Off
    PORTC.6=0; //Left Motor Off
    delay_ms(waitingTime);
    }
}
void moveRight(int gear) {
  PORTC.7=1; //Right Motor Off
  PORTC.6=0; //Left Motor On
  delay_ms(100*gear);
}
```

```
void moveLeft(int gear) {
   PORTC.7=0; //Right Motor Off
   PORTC.6=1; //Left Motor On
   delay_ms(100*gear);
}
//UltraSound Right
    int ultraSoundRight() {
       unsigned short int tempValue , receivedValue;
       PORTC.0 = 1; // Trigger for 10 uSec
       delay_us(15);
       PORTC.0 = 0;
       delay_us(100);
       tempValue = 10;  // for waiting rising edge of echo
       while((PINC.1 == 0)&&(tempValue <=2000))
       {
         delay_us(1);
         tempValue = tempValue+1;
       }
       receivedValue = 10;
       while((PINC.1 == 1)&&(receivedValue <=60000))
       {
         delay_us(1);
         receivedValue = receivedValue+1;
       }
```

```
        receivedValue=receivedValue/22;
        return receivedValue;
       }
//UltraSound Left
     int ultraSoundLeft() {
        unsigned short int tempValue , receivedValue;
        PORTC.2 = 1; // Trigger for 10 uSec
        delay_us(15);
        PORTC.2 = 0;
        delay_us(100);
        tempValue = 10;  // for waiting rising edge of echo
        while((PINC.3 == 0)&&(tempValue <=2000))
        {
          delay_us(1);
          tempValue = tempValue+1;
        }
        receivedValue = 10;
        while((PINC.3 == 1)&&(receivedValue <=60000))
        {
          delay_us(1);
          receivedValue = receivedValue+1;
        }
        receivedValue=receivedValue/22;
        return receivedValue;
```

```
}

void main(void)

{

// Local variables

unsigned short int receivedValueLeftUs, receivedValueRightUs;

int gear=0;

// Input/Output Ports initialization

// Port A initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization

// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out

DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);

// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0

PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

// Port C initialization

// Function: Bit7=Out Bit6=Out Bit5=In Bit4=In Bit3=In Bit2=Out Bit1=In Bit0=Out

DDRC=(1<<DDC7) | (1<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (1<<DDC2) | (0<<DDC1) | (1<<DDC0);
```

```
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=0

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) |
(0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) |
(0<<DDD2) | (0<<DDD1) | (0<<DDD0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) |
(0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) |
(0<<CS02) | (0<<CS01) | (0<<CS00);

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer1 Stopped

// Mode: Normal top=0xFFFF

// OC1A output: Disconnected

// OC1B output: Disconnected

// Noise Canceler: Off
```

```
// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) |
(0<<WGM11) | (0<<WGM10);

TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) |
(0<<CS12) | (0<<CS11) | (0<<CS10);

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: Timer2 Stopped

// Mode: Normal top=0xFF

// OC2 output: Disconnected

ASSR=0<<AS2;

TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) |
(0<<CS21) | (0<<CS20);
```

TCNT2=0x00;

OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);

MCUCSR=(0<<ISC2);

// USART initialization

// USART disabled

UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

// Analog Comparator initialization

// Analog Comparator: Off

// The Analog Comparator's positive input is

// connected to the AIN0 pin

// The Analog Comparator's negative input is

// connected to the AIN1 pin

ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);

SFIOR=(0<<ACME);

// ADC initialization

```
// ADC disabled

ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) |
(0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization

// SPI disabled

SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) |
(0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization

// TWI disabled

TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

gear=5;

   moveForward(gear);  //Dc Motor Speed 5 --> Max Speed

   moveRight(2) ;

   moveLeft(2);

while (1)

     {

     receivedValueRightUs  = ultraSoundRight();

     receivedValueLeftUs   = ultraSoundLeft();

     if (receivedValueRightUs<60 || receivedValueLeftUs<60) {

          if (receivedValueRightUs>=receivedValueLeftUs) {

                moveRight(3);

          }

          else if (receivedValueLeftUs>receivedValueRightUs) {

                moveLeft(3);

          }
```

```
            else {
            moveForward(5);
            }
       }
       else {
       moveForward(5);
       }
   }
}
```

Fire Detection Sensors Data Acquisition

```
#include <mega32a.h>
// Voltage Reference: AVCC pin
#define ADC_VREF_TYPE ((0<<REFS1) | (1<<REFS0) | (0<<ADLAR))
// Standard Input/Output functions
#include <stdio.h>
#include <delay.h>
#include <string.h>
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | ADC_VREF_TYPE;
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
```

```
ADCSRA|=(1<<ADSC);

// Wait for the AD conversion to complete

while ((ADCSRA & (1<<ADIF))==0);

ADCSRA|=(1<<ADIF);

return ADCW;

}

void main(void)

{

// Local variables

int i=0;

char data[100];

int flameSensorValue=0, temperatureSensorValue=0, smokeSensorValue=0;

// Input/Output Ports initialization

// Port A initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) |
(0<<DDA2) | (0<<DDA1) | (0<<DDA0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) |
(0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

//DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) |
(0<<DDB2) | (0<<DDB1) | (0<<DDB0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
```

```
//PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

PORTB=0x00;

DDRB=0xD4;

// Port C initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=in Bit0=In

DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=Out Bit0=In

DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (1<<DDD1) | (0<<DDD0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=0 Bit0=T

PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);

TCNT0=0x00;

OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Disconnected
// OC1B output: Disconnected
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
// Timer/Counter 2 initialization
```

```
// Clock source: System Clock

// Clock value: Timer2 Stopped

// Mode: Normal top=0xFF

// OC2 output: Disconnected

ASSR=0<<AS2;

TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);

TCNT2=0x00;

OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);

MCUCSR=(0<<ISC2);

// USART initialization

// Communication Parameters: 8 Data, 1 Stop, No Parity

// USART Receiver: On

// USART Transmitter: On

// USART Mode: Asynchronous

// USART Baud Rate: 9600
```

```
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (0<<UPE) | (0<<U2X) | (0<<MPCM);

UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);

UBRRH=0x00;

UBRRL=0x33;

// Analog Comparator initialization

// Analog Comparator: Off

// The Analog Comparator's positive input is

// connected to the AIN0 pin

// The Analog Comparator's negative input is

// connected to the AIN1 pin

ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);

SFIOR=(0<<ACME);

// ADC initialization

// ADC Clock frequency: 125,000 kHz

// ADC Voltage Reference: AVCC pin

// ADC Auto Trigger Source: Free Running

ADMUX=ADC_VREF_TYPE;

ADCSRA=(1<<ADEN) | (0<<ADSC) | (1<<ADATE) | (0<<ADIF) | (0<<ADIE) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);

SFIOR=(0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);

// SPI initialization
```

```
// SPI disabled

SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization

// TWI disabled

TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

while (1)

    {

    // Acquisition Sensor Data

    //Flame Sensor Values

    flameSensorValue = read_adc(0); // Read Analog Input

    flameSensorValue = 1024-flameSensorValue; // Read Analog Input and Reverse Input Data

    flameSensorValue = flameSensorValue / 10 ; // Converting to Percentage of Flame Sensor Value

    delay_ms(10);

    //Temperature Sensor Values

    temperatureSensorValue = read_adc(1); // Read Analog Input

    temperatureSensorValue = (5.0 * temperatureSensorValue * 100.0) / 1024;

    delay_ms(10);

    //Smoke Sensor Values

    smokeSensorValue = read_adc(2); // Read Analog Input

SmokeSensorValue = smokeSensorValue*2828/1000; // for converting ppm between 100- 10000ppm

    delay_ms(10);

    //Sending Data via RS232 to Netbook
```

```
sprintf (data, "TemperatureSensor:%d FlameSensor:Per%d SmokeSensor:%d ppm", temperatureSensorValue, flameSensorValue, smokeSensorValue);

      for(i = 0; i<strlen(data); i++)
       {
       putchar(data[i]);
       delay_ms(10);
       }
      putchar(13);  //For Newline
    delay_ms(1000);
   }
}
```

Fire Detection Sensors Data Fusion Algorithm

```
#include <mega32a.h>
// Declare your global variables here
// Voltage Reference: AVCC pin
#define ADC_VREF_TYPE ((0<<REFS1) | (1<<REFS0) | (0<<ADLAR))
// Standard Input/Output functions
#include <stdio.h>
#include <delay.h>
#include <string.h>
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | ADC_VREF_TYPE;
```

```
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=(1<<ADSC);
// Wait for the AD conversion to complete
while ((ADCSRA & (1<<ADIF))==0);
ADCSRA|=(1<<ADIF);
return ADCW;
}
void intializeOutputs()
    {
        int i=0;
        for (i=1;i<15;i++) {
        if (i%3==1) {
           PORTB.0=1; // Green
           PORTB.1=0; // Yellow
           PORTB.2=0; // Red
           delay_ms(200);
         }
        if (i%3==2) {
           PORTB.0=0; // Green
           PORTB.1=1; // Yellow
           PORTB.2=0; // Red
          delay_ms(200);
```

```
      }
     if (i%3==0) {
         PORTB.0=0; // Green
         PORTB.1=0; // Yellow
         PORTB.2=1; // Red
         delay_ms(200);
     }
   }
         PORTB.0=1; // Green
         PORTB.1=1; // Yellow
         PORTB.2=1; // Red
         delay_ms(1500);
         PORTB.0=0; // Green
         PORTB.1=0; // Yellow
         PORTB.2=0; // Red
         delay_ms(500);
         PORTB.0=1; // Green
  }
void main(void)
{
// Declare your local variables here
float firePotential = 0.0;  // 0--> Thereis no fire 0.5--> There is the possibility of fire 1 --> There is fire
float weightTemperatureSensor    =  0.1698957716;
```

```
float weightFlameSensor = 0.7859015343;

float weightSmokeSensor = 0.0707639493;

int flameSensorValue=0, temperatureSensorValue=0, smokeSensorValue=0;

// Input/Output Ports initialization

// Port A initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) |
(0<<DDA2) | (0<<DDA1) | (0<<DDA0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) |
(0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=Out Bit1=Out
Bit0=Out

DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) |
(1<<DDB2) | (1<<DDB1) | (1<<DDB0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) |
(0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

// Port C initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=in Bit0=In

DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) |
(0<<DDC2) | (0<<DDC1) | (0<<DDC0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) |
(0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization
```

```
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=Out Bit0=In

DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (1<<DDD1) | (0<<DDD0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=0 Bit0=T

PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer1 Stopped

// Mode: Normal top=0xFFFF

// OC1A output: Disconnected

// OC1B output: Disconnected

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off
```

```
// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);

TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: Timer2 Stopped

// Mode: Normal top=0xFF

// OC2 output: Disconnected

ASSR=0<<AS2;

TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);

TCNT2=0x00;

OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);

MCUCSR=(0<<ISC2);

// USART initialization

// Communication Parameters: 8 Data, 1 Stop, No Parity

// USART Receiver: On

// USART Transmitter: On

// USART Mode: Asynchronous

// USART Baud Rate: 9600

UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (0<<UPE) | (0<<U2X) | (0<<MPCM);

UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);

UBRRH=0x00;

UBRRL=0x33;

// Analog Comparator initialization

// Analog Comparator: Off

// The Analog Comparator's positive input is

// connected to the AIN0 pin
```

```
// The Analog Comparator's negative input is

// connected to the AIN1 pin

ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);

SFIOR=(0<<ACME);

// ADC initialization

// ADC Clock frequency: 125,000 kHz

// ADC Voltage Reference: AVCC pin

// ADC Auto Trigger Source: Free Running

ADMUX=ADC_VREF_TYPE;

ADCSRA=(1<<ADEN) | (0<<ADSC) | (1<<ADATE) | (0<<ADIF) | (0<<ADIE) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);

SFIOR=(0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);

// SPI initialization

// SPI disabled

SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization

// TWI disabled

TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

intializeOutputs();

while (1)

    {

    // Place your code here

    //Flame Sensor
```

```
flameSensorValue = read_adc(0); // Read Analog Input

flameSensorValue = 1024-flameSensorValue; // Read Analog Input and Reverse Input Data

flameSensorValue = flameSensorValue / 10 ; // Converting to Percentage of Flame Sensor Value

delay_ms(10);

//Temperature Sensor

temperatureSensorValue = read_adc(1); // Read Analog Input

temperatureSensorValue = (5.0 * temperatureSensorValue * 100.0) / 1024;

delay_ms(10);

//Smoke Sensor

smokeSensorValue = read_adc(2); // Read Analog Input

smokeSensorValue = smokeSensorValue*100; // for converting ppm between 100-10000ppm

delay_ms(10);

// Limiting results between 0 and 1

flameSensorValue          =  flameSensorValue/100;

temperatureSensorValue  =  temperatureSensorValue/150;

smokeSensorValue          =  smokeSensorValue/1000;

// Formula

firePotential     =  flameSensorValue*weightFlameSensor;

firePotential    +=  temperatureSensorValue*weightTemperatureSensor;

firePotential    +=  smokeSensorValue*weightSmokeSensor;

   if  (firePotential<0.30) {  //There is no Fire

      PORTB.0=1; // Green
```

```
            PORTB.1=0; // Yellow

            PORTB.2=0; // Red

        }else if  (firePotential>=0.30 && firePotential<0.60) {   //There is the possibility of fire

            PORTB.0=0; // Green

            PORTB.1=1; // Yellow

            PORTB.2=0; // Red

        }else if  (firePotential>=0.60 && firePotential<=1.00) {   //There is the possibility of fire

            PORTB.0=0; // Green

            PORTB.1=0; // Yellow

            PORTB.2=1; // Red

        }

    }
```

Environment Scanning With Servo Motor

```
#include <mega32a.h>

#include <delay.h>

void main(void)

{

// Local variables

float i=0;

// Input/Output Ports initialization

// Port A initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
```

```
DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

// Port B initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

// Port C initialization

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization

// Function: Bit7=In Bit6=Out Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In

DDRD=(0<<DDD7) | (1<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);

// State: Bit7=T Bit6=0 Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

// Timer/Counter 0 initialization

// Clock source: System Clock
```

```
// Clock value: Timer 0 Stopped

// Mode: Normal top=0xFF

// OC0 output: Disconnected

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer1 Stopped

// Mode: Normal top=0xFFFF

// OC1A output: Disconnected

// OC1B output: Disconnected

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);

TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);

TCNT1H=0x00;

TCNT1L=0x00;
```

```
ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

// Clock source: System Clock

// Clock value: Timer2 Stopped

// Mode: Normal top=0xFF

// OC2 output: Disconnected

ASSR=0<<AS2;

TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);

TCNT2=0x00;

OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);

MCUCSR=(0<<ISC2);
```

```
// USART initialization

// USART disabled

UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) |
(0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

// Analog Comparator initialization

// Analog Comparator: Off

// The Analog Comparator's positive input is

// connected to the AIN0 pin

// The Analog Comparator's negative input is

// connected to the AIN1 pin

ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) |
(0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);

SFIOR=(0<<ACME);

// ADC initialization

// ADC disabled

ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) |
(0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization

// SPI disabled

SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) |
(0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization

// TWI disabled

TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

PORTD.6=0;

delay_ms(1000);
```

```
while (1)
    {
    // For scanning enviroment with flame sensor
    for (i=3.40;i>0.30;i-=0.01){
    PORTD.6=0;
    PORTD.6=1;
    delay_ms(i);
    PORTD.6=0;
    delay_ms(20);
    if (i>1.5 && i<1.7)   {
    delay_ms(20);
    }
    }
    //delay_ms(300);
    for (i=0.10;i<3.90;i+=0.01){
    PORTD.6=0;
    PORTD.6=1;
    delay_ms(i);
    PORTD.6=0;
    delay_ms(20);
    if (i>1.5 && i<1.7)   {
    delay_ms(20);
    }
```

Appendix-2 (Data Acquisition and Control Circuits)

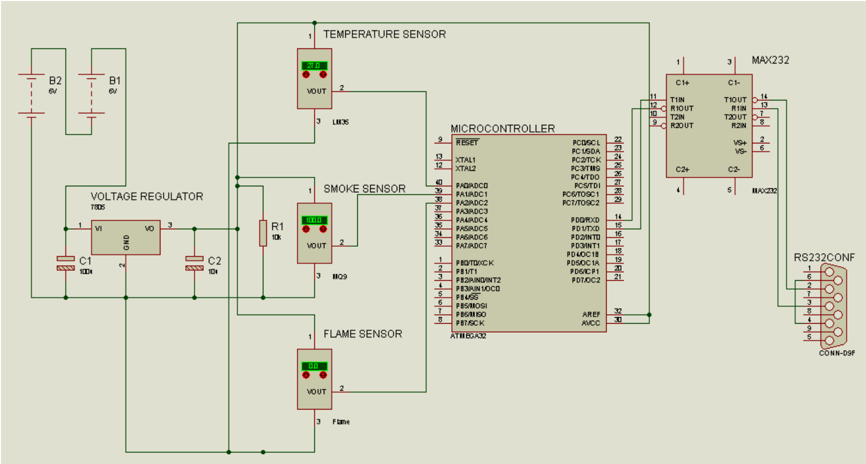


Figure 7.1. Data acquisition and communication circuit

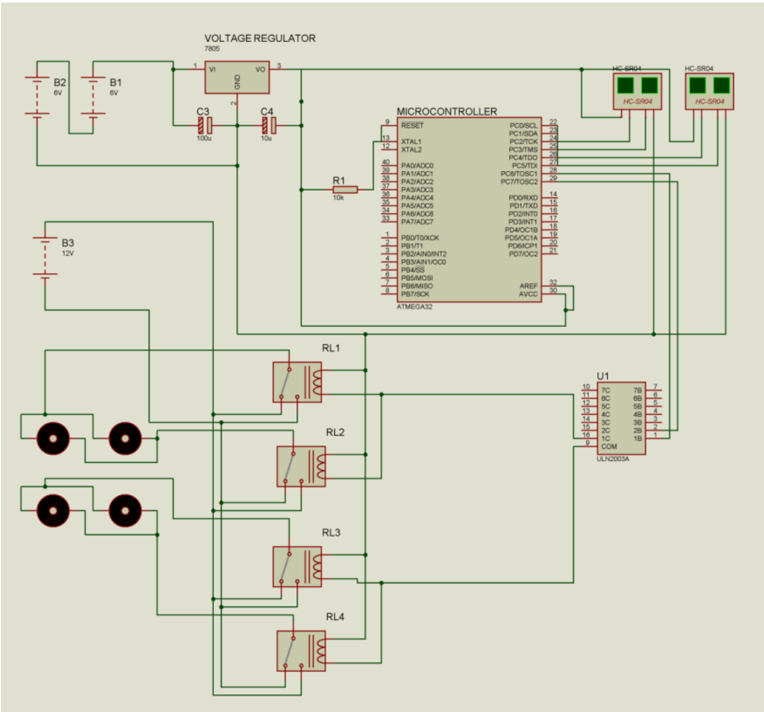


Figure 7.2. DC motor control circuit

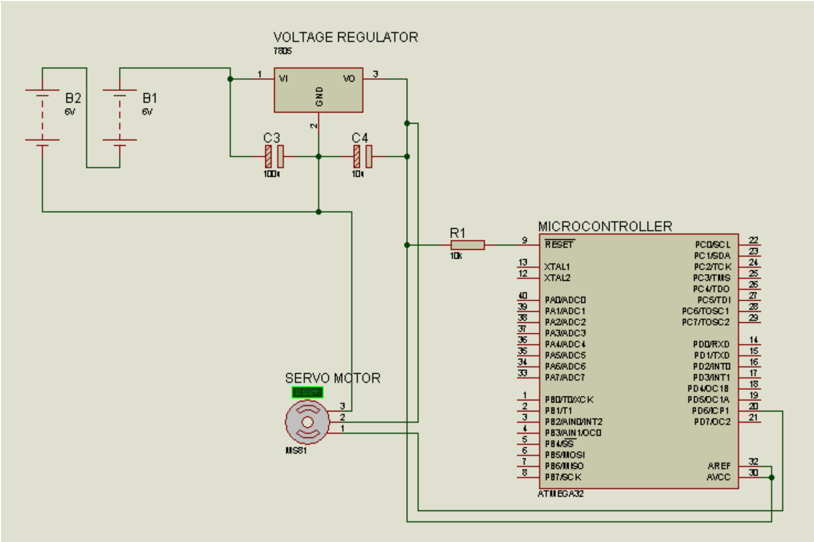


Figure 7.3. Servo motor control circuit

Appendix-3 (Force and Torque Calculator)

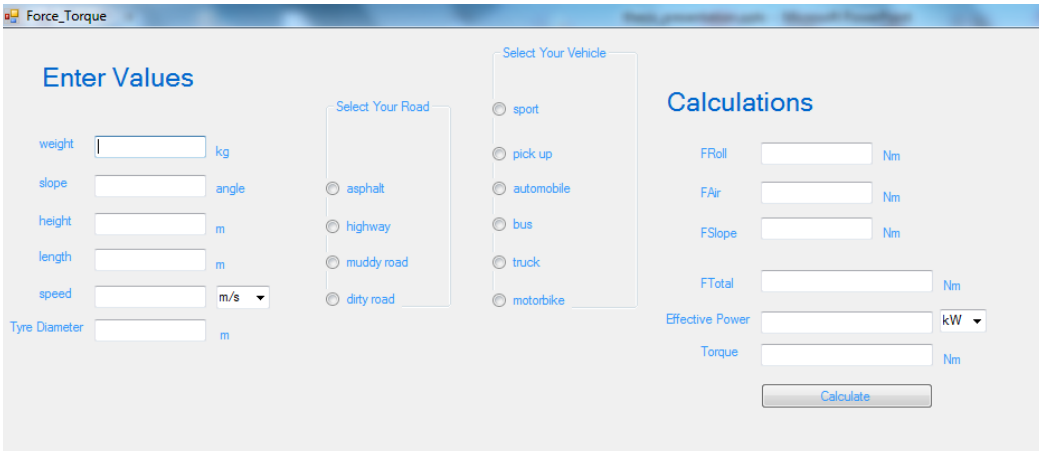


Figure 7.4. Force-Torque Calculator

RESUME

Hilmi Saygın SUCUOĞLU was born in İzmir, in 1987. He completed his high school at İzmir Konak Anatolian High School, in June 2005. He received his Bachelor of Science degree from Uşak University, Mechanical Engineering Department in 2009. In 2012, he was accepted to Adnan Menderes University Mechanical Engineering Department for Master of Science in engineering programme. He has been working as a Research Assistant at Adnan Menderes University since 2013.