ROBOT COMMUNICATION AND TARGET TRACKING SYSTEM WITH CONTROLLER DESIGN AND IMPLEMENTATION USING ARDUINO

A Project Report

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

BY

C.SURYA PRAKASH REDDY	**-169L1A0408**
B. LOKESWARA	**-169L1A0404**
K.S. MANIKANTA	**-169L1A0423**
R. ANIL KUMAR REDDY	**-169L1A0450**
V.VENKATA PAVAN REDDY	**-169L1A0464**

Under the esteemed guidance of

Mr. T. VENKATA RAMANA, MTech, MISTE
Assistant Professor, Department of ECE

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SIDDARTHA EDUCATIONAL ACADEMY GROUP OF INSTITUTIONS

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)

Near C. Gollapalli, Tirupati - 517 505

2016 – 2020

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SIDDARTHA EDUCATIONAL ACADEMY GROUP OF INSTITUTIONS
Near C. Gollapalli, Tirupati - 517 505

CERTIFICATE

This is to certify that the project report entitled "**ROBOT COMMUNICATION AND TARGET TRACKING SYSTEM WITH CONTROLLER DESIGN AND IMPLEMENTATION USING ARDUINO**" is a bonafide work carried out by

C.SURYA PRAKASH REDDY	**-169L1A0408**
B. LOKESWARA	**-169L1A0404**
K.S. MANIKANTA	**-169L1A0423**
R. ANIL KUMAR REDDY	**-169L1A0450**
V.VENKATA PAVAN REDDY	**-169L1A0464**

B. Tech students of Siddartha Educational Academy Group of Institutions, Affiliated to JNTUA, Ananthapuramu in partial fulfillment of the requirements for the award of the Degree of **BACHELOR OF TECHNOLOGY** with the specialization in **ELECTRONICS AND COMMUNICATION ENGINEERING** during the Academic year2016-2020.

GUIDE	**HEAD OF THE DEPARTMENT**
Mr. T. VENKATA RAMANA	**Dr. K. PURUSHOTHAM PRASAD,**
Assistant Professor,	**Professor,**
Department of ECE, SEAGI.	**Department of ECE, SEAGI.**

Viva-Voice Conducted on___________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SIDDARTHA EDUCATIONAL ACADEMY GROUP OF INSTITUTIONS
Near C. Gollapalli, Tirupati - 517 505

DECLARATION

We hereby declare that the project report work entitled "**ROBOT COMMUNICATION AND TARGET TRACKING SYSTEM WITH CONTROLLER DESIGN AND IMPLEMENTATION USING ARDUINO**" is entirely our original work carried out under the guidance of **Mr. T.VENKATA RAMANA,** Assistant Professor, Department Electronics and Communication Engineering, Siddartha Educational Academy Group of Institutions, C. Gollapalli, Tirupati, Affiliated to JNTUA, Ananthapuramu, for the award of the degree of **BACHELOR OF TECHNOLOGY** with the specialization in **ELECTRONICS AND COMMUNICATIONENGINEERING**. The seminar report has not been submitted in a part or full for the award of any degree or diploma of this or any other university or institute.

C.SURYA PRAKASH REDDY	**-169L1A0408**
B. LOKESWARA	**-169L1A0404**
K.S. MANIKANTA	**-169L1A0423**
R. ANIL KUMAR REDDY	**-169L1A0450**
V.VENKATA PAVAN REDDY	**-169L1A0464**

ACKNOWLEDGEMENT

All endeavors over a long period can be successful only with the advice and support of many well-wishers. We take this opportunity to express my gratitude and appreciation to all of them.

we wish to express deep sense of gratitude to my beloved and respected guide **Mr. T.VENKATA RAMANA**, Assistant Professor, Department of Electronics and Communication Engineering, Siddartha Educational Academy Group of Institutions, Tirupati, for his valuable guidance, suggestions and keen interest enriched throughout the course of project work and encouragement given to me for the successful completion of work.

Our sincere thanks to **Dr. K. PURUSHOTHAM PRASAD Head,** Department of Electronics and Communication Engineering for his valuable advice, guidance and encouragement given to us for the successful completion of this work.

we extend sincere thanks to the **Dr. K. RAJASEKHAR, Principal** for his kind co- operation in completing and making the seminar a success.

we would like to thank the **Management** for their kind co-operation and for providing infrastructure facilities.

we extend our thanks to all the **Teaching staff and Non-Teaching staff** of the Department of Electronics and Communication Engineering for their support and encouragement during the course of my project work. we also thank **My Parents** for being helpful in many ways in successful completion of my work.

Finally, we thank all those who helped us directly or indirectly in successful completion of this seminar work.

C.SURYA PRAKASH REDDY -169L1A0408

B. LOKESWARA -169L1A0404

K.S. MANIKANTA -169L1A0423

R. ANIL KUMAR REDDY -169L1A0450

V.VENKATA PAVAN REDDY -169L1A0464

ABSTRACT

Target detection and tracking encompasses a variety of decisional problems such as coverage, surveillance, search, patrolling, observing and pursuit-evasion along with others. These problems are studied by several communities that tackle them using diverse formulations, hypotheses and approaches. This variety of target related robotics problems are pertinent for a large spectrum of applications has motivated a large amount of contributions, which have mostly been surveyed according to one or another viewpoint. In this project our objective is to go beyond the frontiers of specific communities and specific problems, and to enlarge the scope of prior surveys. We define classes of missions and problems and relate the results from various communities according to a unifying taxonomy. We review various work related to each class of problems identified in the taxonomy, highlighting the different approaches, models and results. Finally, we propose a transverse synthesis which analyzes the approaches, models and lacks that are recurrent through all the tackled problems, and isolate the current main research directions. As we can see, now a day's lot of soldiers were sacrificing their lives for the sake of our country and people and also while occurring natural calamities it is very difficult to perform rescue operations for the rescue team. To overcome these problems, we use this robotic communication system.

By keeping these robots at the border these not only keep an eye on the movement of terrorists but also give the information if any unusual thing happens at the boarders. While coming to the rescue operations during floods, earthquake etc.… robots are useful to capture the situation with its resolution camera ,and it can go to the places where humans cannot go, this can be possible with its small size and helps to perform rescue operations and helps to find a better way for quick response. This robot can be control by using IOT networks based on the instructions given by the user, this robot performs several tasks.

INDEX

CHAPTER NUMBER	TITLE	PAGE NUMBER
	CERTIFICATE	
	DECLARATION	
	ACKNOWLEDGEMENT	
	ABSTRACT	
	CONTENTS	i
	LIST OF FIGURES	iii
	LIST OF TABLES	v
	ABBREVATIONS	vi
CHAPTER-1	**EMBEDDED SYSTEMS**	**(1-8)**
	1.1 INTRODUCTION TO EMBEDDED SYSTEM	1
	1.2 HISTORY	4
	1.3 CHARACTERISTICS OF EMBEDDED SYSTEM	4
	1.3.1 USER INTERFACES	6
	1.3.2 PERIPHERALS	6
	1.3.3 DEBUGGING	6
	1.4 MICROCONTROLLER VERSUS MICROPROCESSOR	7
	1.5 MICROCONTROLLERS FOR EMBEDDED SYSTEMS	8
CHAPTER-2	**ROBOTICS**	**(9-12)**
	2.1 INTRODUCTION	9
	2.2 ETYMOLOGY	10
	2.3 HISTORY	10
	2.4 APPLICATIONS	11
CHAPTER-3	**INTRODUCTION TO PROJECT**	**(13-20)**
	3.1 MOTIVATION	13
	3.2 TARGETDETECTION	14
	3.3 TARGETTRACKING	16
	3.4 BLOCK DIAGRAM	18
	3.5 TECHNICAL SPECIFICATIONS	19
	3.5.1 HARDWARE	19
	3.5.2 SOFTWARE	19

	3.6 WORKING	19
	3.7 APPLICATIONS	20
CHAPTER-4	**MICROCONTROLLER**	**(21-30)**
	4.1 INTRODUCTION	21
	4.1.1 ARDUINO UNO MICROCONTROLLER	21
	4.2 ARDUINO UNO BOARD	24
	4.2.1 TECHNICAL SPECIFICATIONS	24
	4.2.2 PIN DESCRIPTION	27
CHAPTER-5	**HARDWARE COMPONENTS**	**(31-59)**
	5.1 DC BATTERY	31
	5.2 BLUETOOTH	33
	5.3 IR SENSOR TECHNOLOGY	36
	5.3.1 IR GENERATION	36
	5.4 IR OBSTACLE SENSOR	37
	5.5 ULTRASONIC OBSTACLE SENSOR:	39
	5.6 L293D- CURRENT DRIVER	42
	5.7 DC MOTORS	46
	5.7.1 TYPES OF MOTORS	50
	5.7.2. HORSEPOWER & RPM	50
	5.8 LASER DIODE(650NM)	55
	5.8.1 CHARACTERISTICS CURVES	56
CHAPTER-6	**ARDUINO IDE**	**(60-73)**
	6.1 INTRODUCTION	60
	6.1.1 ARDUINO DATA TYPES	60
	6.1.2 ARDUINO PROGRAMMING STRUCTURE	71
CHAPTER-7	**RESULT AND ANALYSIS**	**(74-77)**
CHAPTER-8	**CONCLUSION AND FUTURE SCOPE**	**(78-80)**
	8.1 CONCLUSION	78
	8.2 FUTURE SCOPE	78
	REFERENCES	**81**

LIST OF FIGURES

FIGURE NUMBER	NAME OF THE FIGURE	PAGE NUMBER
1.1	EMBEDDED SYSTEM DESIGN CELLS	2
1.2	V DIAGRAM	3
3.1	PROPOSED TAXONOMY OF THE TARGET MANAGEMENT PROBLEMS	14
3.2	BLOCK DIAGRAM OF ROBOT	18
4.1	ARDUINO UNO BOARD	24
4.2	PIN DIAGRAM OF ATMEGA MICROCONTROLLER	27
5.1	DIFFERENT SHAPES AND SIZES AND VOLTAGES OF BATTERIES	32
5.2	HC-05 BLUETOOTH MODULE	33
5.3	INFRARED SCALE	36
5.4	PULSAT0ING 36KHZ INFRARED	36
5.5	RECEIVER'S OUTPUT	37
5.6	IR OBSTACLE SENSOR	37
5.7	IR OBSTACLE SENSOR MODULES	38
5.8	ULTRASONIC SENSOR HC SR04	39
5.9	WORKING DIAGRAM	41
5.10	L293D DRIVE	42
5.11	LOGIC DIAGRAM	44
5.12	INTERFACING WITH AURDINO	45
5.13	A LEFT-HAND RULE	46
5.14	FUNCTIONAL DIAGRAM MOTOR	47
5.15	DC MOTO FIELD DIAGRAM IN DIFFERENT VIEW	48
5.16	DC MOTO FIELD DIAGRAM PARTS TOGETHER	49
5.17	SINGLE PHASE	52
5.18	THREE PHASE	52
5.19	WORM GEARS	53
5.20	LASER DIODE(650NM)	55
5.21	2PIN LASER DIODE PINOUT	55
5.22	EFFECT OF TEMPERATURE	57

5.23	LASER BEAM DIVERGENCE IN PARALLEL AND PERPENDICULAR PLANE	57
5.24	FORWARD VOLTAGE V/S FORWARD CURRENT	57
5.25	OUTPUT POWER V/S FORWARD CURRENT	58
6.1	USB CABLE	65
6.2	DOWNLOADING AND SAVING THE ARDUINO IDE	65
6.3	LAUNCHING OF ARDUINO IDE	66
6.4	CREATING A NEW PROJECT	67
6.5	OPENING OF EXISTING PROJECTS	68
6.6	SELECTING ARDUINO UNO BOARD	69
6.7	SELECTING SERIAL PORT COMMUNICATION	70
6.8	DESCRIPTION OF ICON ON NEW PROJECT WINDOW	70
6.9	STARTING WINDOW OF NEW PROJECT	72
7.1	PROPOSED ROBOT MODEL	74
7.2	HC-05 BLUETOOTH CONTROL PANNEL AND HC-05 DEVICE	75
7.3	C IPWEB APPLICATION AND INTERFACE	76
7.4	IPWEBCAM DISPLAY ON BROWSER	76

LIST OF TABLES

TABLE NUMBER	TITLE	PAGE NUMBER
5.1	BLUETOOTH SPECIFICATIONS	35
5.2	ULTRASONIC SENSOR PIN CONFIGURATION	40
5.3	FUNCTIONAL TABLE	44
5.4	CONTROLLING DIRECTIONS OF ROBOT	45
5.5	HORSE POWER RATING COMPARED TO THE AMPERE RATING	51
5.6	OPERATIONAL SPECIFICATIONS	54

ABBRIVATIONS

DSP	–	DIGITAL SIGNAL PROCESSING
UNIX	–	UNIPLEXED INFORMATION AND COMPUTER SYSTEMS
OS	–	OPERATING SYSTEM
ICS	–	INTEGRATED CIRCUITS.
CPU	–	CENTRAL PROCESSING UNIT
LED	–	LIGHT EMITTING DIODE
LCDS	–	LIQUIDE CRYSTAL DISPLAY
SCI	–	SERIAL COMMUNICATIONS INTERFACES
I2C	–	INTER-INTEGRATED CIRCUIT
SPI	–	SERIAL PERIPHERAL INTERFACE
SSC	–	SYNCHRONOUS SERIAL COMMUNICATION
ESSI	–	ENHANCED SYNCHRONOUS SERIAL INTERFACE
USB	–	UNIVERSAL SERIAL BUS
CAN-BUS	–	CONTROLLER AREA NETWORK BUS
LIN-BUS	–	LOCAL INTERCONNECT NETWORK BUS
PROFIBUS	–	PROCESS FIELD BUS
PLL	–	PHASE LOCK LOOP
GPIO	–	GENERAL PURPOSE INPUT / OUTPUT
ADC/DAC	–	ANALOG TO DIGITAL OR DIGITAL TO ANALOG CONVERTERS
JTAG	–	JOINT TEST ACTION GROUP
ISP	–	IN-SYSTEM PROGRAMMING
ICSP	–	IN-CIRCUIT SERIAL PROGRAMMING
BDM PORT	–	BACKGROUND DEBUG MODE PORT
ICD	–	IN-CIRCUIT DEBUGGER
ICE	–	IN- CIRCUIT EMULATOR
RAM	–	RANDOM ACCESS MEMORY
ROM	–	READ ONLY MEMORY
PWM	–	PULSE WIDTH MODULATION
EEPROM	–	ELECTRICALLY ERASABLE PROGRAMMABLE READ-ONLY MEMORY

TTL	–	TRANSISTOR–TRANSISTOR LOGIC
SS	–	SLAVE SELECT
MOSI	–	MASTER OUT SLAVE IN
MISO	–	MASTER IN SLAVE OUT
SCK	–	SERIAL CLOCK
UART	–	UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER
ARM	–	ADVANCED RISC MACHINE
CCW	–	COUNTER-CLOCKWISE
CW	–	CLOCKWISE
COD	–	CATASTROPHIC OPTICAL DAMAGE
IDE	–	INTEGRATED DEVELOPMENT ENVIRONMENT
IOT	–	INTERNET OF THINGS

CHAPTER-1

EMBEDDED SYSTEMS

1.1 INTRODUCTION TO EMBEDDED SYSTEM

An Embedded system is a computer system designed for specific control functions within a larger system and often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

Embedded systems contain processing cores that are typically either microcontrollers or digital signal processors (DSP). The key characteristic, however, is being dedicated to handle a particular task. They may require very powerful processors and extensive communication, for example air traffic control systems may usefully be viewed as embedded, even though they involve mainframe computers and dedicated regional and national networks between airports and radar sites (each radar probably includes one or more embedded systems of its own).

Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically, embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not a strictly definable term, as most systems have some element of extensibility or programmability. For example, handheld computers share some elements with embedded systems such as the operating systems and microprocessors that power them, but they allow different applications to be loaded and peripherals to be connected. Moreover, even systems that do not expose programmability as a primary feature generally need to support software updates. On a continuum from "general purpose" to "embedded", large application systems will have subcomponents at most points even if the system as a whole is "designed to perform one or a few dedicated functions", and is thus appropriate to call ‘embedded’.

An embedded system is not a computer system that is used primarily for processing, not a software system on PC or UNIX, not a traditional business or scientific application as shown in figure1.1. High-end embedded & lower end embedded systems, High-end embedded system - Generally 32, 64 Bit Controllers used with OS. Examples Personal Digital Assistant and Mobile phones etc. Lower end embedded systems - Generally 8,16 Bit Controllers used with a minimal operating systems and hardware layout designed for the specific purpose that as shown in figure 1.2.

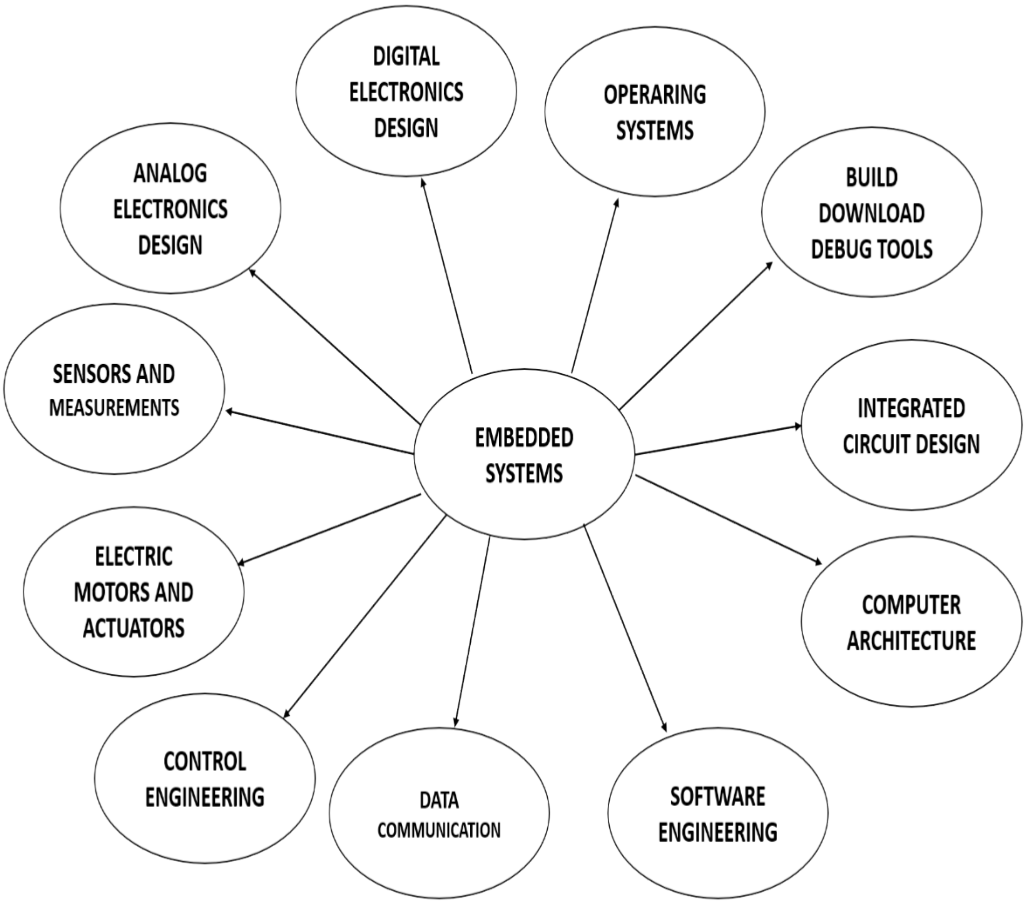


Figure 1.1 Embedded system design cells

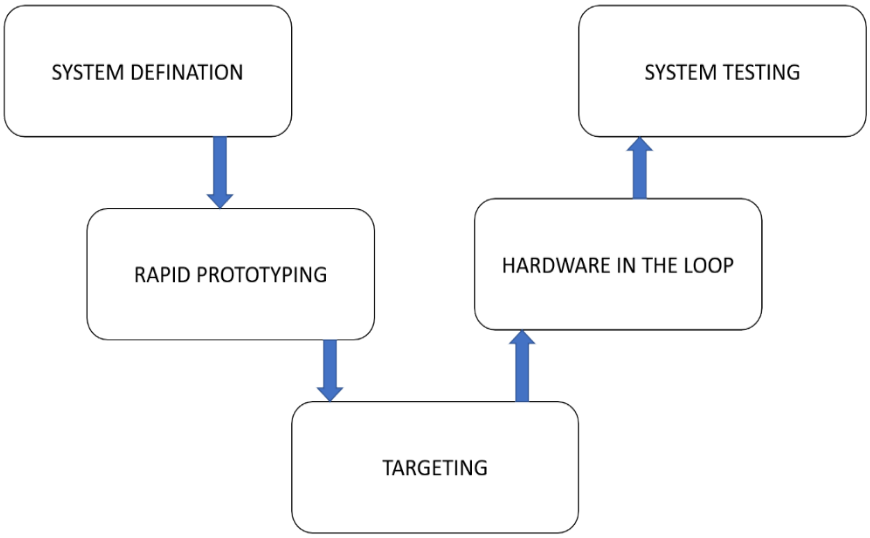


Figure 1.2 V Diagram

Examples of Embedded Systems:

Embedded systems are found in wide range of application areas. Originally they were used only for expensive industrial control applications, but as technology brought down the cost of dedicated processors, they began to appear in moderately expensive applications such as automobiles, communication and office equipment and television Today's embedded systems are so inexpensive that they are used in almost every electronic product in our life. Embedded systems are often designed for mass production.

Some examples of embedded systems:

1. Automatic Teller Machines
2. Cellular telephone and telephone switches
3. Computer network equipment
4. Computer printers
5. Disk drives
6. Engine controllers and antilock brake controllers for automobiles
7. Home automation products
8. Handheld calculators
9. Household appliances
10. Medical equipment

11. Measurement equipment
12. Multifunction wrist watches
13. Multifunction printers

1.2 HISTORY

One of the first recognizably modern embedded systems was the Apollo Guidance Computer, developed by “Charles Stark Draper” at the MIT Instrumentation Laboratory. At the project inception, the Apollo guidance computer was considered the riskiest item in the Apollo project as it employed the then newly developed monolithic integrated circuits to reduce the size and weight. An early mass produced embedded system was the Automatics D-17 guidance computer for the Minuteman missile, released in 1961.It was built from transistor logic and had a hard disk for main memory .When the Minuteman II went into production in 1966, the D-17 was replaced with a new computer that was the first high volume use of integrated circuits. This program alone reduced prices on quad and gate ICs from $1000/ each to $3/ each permitting their use in commercial products.

Since these early applications in the 1960s, embedded systems have come down in price and there has been a dramatic rise in processing power and functionality. The first microprocessor for example, the Intel 4004, was designed for calculators and other small systems but still required many external memory and support chips. In 1978 National Engineering Manufactures Association released a “standard” for programmable microcontrollers, including almost any computer-based controllers, such as single board computers, numerical and event-based controllers.

Embedded Systems are designed to some specific task, rather than be a general-purpose computer for multitasks. Some also have real-time performances constraints that must be met, for reasons such as safety and usability; others may have low or no performance requirement, allowing the system hardware to be simplified to reduce cost

1.3 CHARACTERISTICS OF EMBEDDED SYSTEM

Embedded systems are designed to do some specific task, rather than be a general-purpose computer for multiple tasks. Some also have real-time performance constraints that must be met, for reasons such as safety and usability; others may have low or no performance requirements, allowing the system hardware to be simplified to reduce costs.

Embedded systems are not always standalone devices. Many embedded systems consist of small, computerized parts within a larger device that serves a more general purpose. For example, the Gibson Robot Guitar features an embedded system for tuning the strings, but the overall purpose of the Robot Guitar is, of course, to play music. Similarly, an embedded system in an automobile provides a specific function as a subsystem of the car itself.

The program instructions written for embedded systems are referred to as firmware, and are stored in read-only memory or Flash memory chips. They run with limited computer hardware resources: little memory, small or non-existent keyboard and/or screen.

An embedded system is any computer system hidden inside a product other than a computer.They will encounter a number of difficulties when writing embedded system software in addition to those we encounter when we write applications.

1. Throughput – Our system may need to handle a lot of data in a short period of time.
2. Response–Our system may need to react to events quickly.
3. Testability–Setting up equipment to test embedded software can be difficult.
4. Debugability–Without a screen or a keyboard, finding out what the software is doing wrong (other than not working) is a troublesome problem.
5. Reliability – embedded systems must be able to handle any situation without human intervention.
6. Memory space – Memory is limited on embedded systems, and you must make the software and the data fit into whatever memory exists.
7. Program installation – you will need special tools to get your software into embedded systems.
8. Power consumption – Portable systems must run on battery power, and the software in these systems must conserve power.
9. Processor hogs – computing that requires large amounts of CPU time can complicate the response problem.

10. Cost – Reducing the cost of the hardware is a concern in many embedded system projects; software often operates on hardware that is barely adequate for the job.

Embedded systems have a microprocessor/ microcontroller and a memory. Some have a serial port or a network connection. They usually do not have keyboards, screens or disk drives

1.3.1 USER INTERFACES

Embedded systems range from no user interface at all - dedicated only to one task - to full user interfaces similar to desktop operating systems in devices such as PDAs.

1.3.2 PERIPHERALS

Embedded systems range from no user interface at all - dedicated only to one task - to complex graphical user interfaces that resemble modern computer desktop operating systems. Simple embedded devices use buttons, LEDs, graphic or character LCDs (for example popular HD 44780LCD) with a simple system.

- Embedded systems talk with outside world via peripherals, such as:
- Serial Communications Interfaces (SCI): RS-232, RS-422, RS-485 etc.
- Synchronous Serial Communication Interface: I2c, SPI, SSC and ESSI (Enhanced -Synchronous Serial Interface)
- Universal Serial Bus (USB)
- Multi Media Cards (SD Cards, Compact Flash etc.)
- Networks: Ethernet, Lon Works etc.
- Field buses: CAN-Bus, LIN-Bus, PROFIBUS etc.
- Timers: PLLS(s), Capture /Compare and Time Processing Units
- Discrete IO: Aka General Purpose Input / output (GPIO)
- Analog to Digital or Digital to Analog (ADC/DAC)
- Debugging: JTAG, ISP, ICSP, BDM Port, BITP and DP9 ports.

1.3.3 DEBUGGING

Embedded debugging may be performed at different levels, depending on the facilities available. From simplest to most sophisticate they can be roughly grouped into the following areas:

- Interactive resident debugging using the simple shell provided by the embedded operating system (e.g. Forth and Basic)
- External debugging using logging or serial port output to trace operating using either a monitor in flash or using a debug server like the Remedy Debugger which even works for heterogeneous multi core systems.
- An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a JTAG or NEXUS interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.
- An in- circuit emulator (ICE) replaces the microprocessor with a simulated equivalent, providing full control over all aspects of the microprocessor.
- A complete emulator provides a simulation of all aspects of the hardware, allowing all of it to be controlled and modified and allowing debugging on a normal PC.

Applications:

1. Military and aerospace embedded software applications
2. Communication Applic29ations, Intelligent, autonomous sensors.
3. Mastering the complexity of applications.
4. Reduction of product design time.
5. Real time processing of ever-increasing amounts of data

1.4 MICROCONTROLLER VERSUS MICROPROCESSOR

What is the difference between a Microprocessor and Microcontroller? By microprocessor is meant the general-purpose Microprocessors such as Intel's X86 family (8086, 80286, 80386, 80486, and the Pentium) or Motorola's 680X0 family (68000, 68010, 68020, 68030, 68040, etc.). These microprocessors contain no RAM, no ROM, and no I/O ports on the chip itself. For this reason, they are commonly referred to as general-purpose Microprocessors.

A system designer using a general-purpose microprocessor such as the Pentium or the 68040 must add RAM, ROM, I/O ports, and timers externally to make them functional. Although the addition of external RAM, ROM, and I/O ports makes these systems bulkier and much more expensive, they have the advantage of versatility such that the designer can decide on the amount of RAM, ROM and I/O ports needed to fit the task at hand. This is not the case with Microcontrollers.

A Microcontroller has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip. In other words, the processor, the RAM, ROM, I/O ports and the timer are all embedded together on one chip; therefore, the designer cannot add any external memory, I/O ports, or timer to it. The fixed amount of on-chip ROM, RAM, and number of I/O ports in Microcontrollers makes them ideal for many applications in which cost and space are critical.

In many applications, for example a TV remote control, there is no need for the computing power of a 486 or even an 8086 microprocessor. These applications most often require some I/O operations to read signals and turn on and off certain bits

1.5 MICROCONTROLLERS FOR EMBEDDED SYSTEMS

In the Literature discussing microprocessors, we often see the term Embedded System. Microprocessors and Microcontrollers are widely used in embedded system products. An embedded system product uses a microprocessor (or Microcontroller) to do one task only. A printer is an example of embedded system since the processor inside it performs one task only; namely getting the data and printing it. Contrast this with a Pentium based PC. A PC can be used for any number of applications such as word processor, print-server, bank teller terminal, Video game, network server, or Internet terminal. Software for a variety of applications can be loaded and run. Of course, the reason a pc can perform myriad tasks is that it has RAM memory and an operating system that loads the application software into RAM memory and lets the CPU run it.

In this robot as the fire sensor senses the fire, it senses the signal to microcontroller. In an Embedded system, there is only one application software that is typically burned into ROM. An x86 PC contains or is connected to various embedded products such as keyboard, printer, modem, disk controller, sound card, CD-ROM drives, mouse, and so on. Each one of these peripherals has a Microcontroller inside it that performs only one task.

CHAPTER-2
ROBOTICS

2.1 INTRODUCTION

Robotics is an interdisciplinary branch of engineering and science that includes mechanical engineering, electronic engineering, information engineering, computer science, and others.

Robotics deals with the design, construction, operation, and use of robots, and computer systems for their control, sensory feedback, and information processing.

These technologies are used to develop machines that can substitute for humans and replicate human actions. Robots can be used in many situations for lots of purposes. today many are used in dangerous environments, manufacturing processes, or where humans cannot survive (e.g. in space, underwater, in high heat, and clean up and containment of hazardous materials and radiation). Robots can take on any form but some are made to resemble humans in appearance. This is said to help in the acceptance of a robot in certain replicative behaviors usually performed by people. Such robots attempt to replicate walking, lifting, speech, cognition, or any other human activities. Many of today's robots are inspired by nature, contributing to the field of bio-inspired robotics.

The concept of creating machines that can operate autonomously dates back to classical times, but research into the functionality and potential uses of robots did not grow substantially until the 20th century. Throughout history, it has been frequently assumed by various scholars, inventors, engineers, and technicians that robots will be able to mimic human behavior and manage tasks in a human-like fashion. Today, robotics is a rapidly growing field, as technological advances continue; researching, designing, and building new robots serve various practical purposes, whether domestically, commercially, or militarily. Many robots are built to do jobs that are hazardous to people, such as defusing bombs, finding survivors in unstable ruins, and exploring mines and shipwrecks. Robotics is also used in STEM (science, technology, engineering, and mathematics) as a teaching aid. The advent of nanorobots, microscopic robots that can be injected into the human body, could revolutionize medicine and human health. Robotics is a branch of engineering that involves the conception, design, manufacture, and operation of robots.

2.2. ETYMOLOGY

The word robotics was derived from the word robot, which was introduced to the public by Czech writer Karel Capek in his play R.U.R. (Rossum's Universal Robots), which was published in 1920. The word robot comes from the Slavic word robata, which means slave/servant. The play begins in a factory that makes artificial people called robots, creatures who can be mistaken for humans – very similar to the modern ideas of androids. Karel Capek himself did not coin the word. He wrote a short letter about etymology in the Oxford English Dictionary in which he named his brother Josef Capek as its actual originator.

According to the Oxford English Dictionary, the word robotics was first used in print by Isaac Asimov, in his science fiction short story "Liar", published in May 1941 in Astounding Science Fiction. Asimov was unaware that he was coining the term; since the science and technology of electrical devices are electronics, he assumed robotics already referred to the science and technology of robots. In some of Asimov's other works, he states that the first use of the word robotics was in his short story Run-around (Astounding Science Fiction, March 1942), where he introduced his concept of The Three Laws of Robotics. However, the original publication of "Liar" predates that of "Run-around" by ten months, so the former is generally cited as the word's origin.

2.3. HISTORY

In 1948, Norbert Wiener formulated the principles of cybernetics, the basis of practical robotics.

Fully autonomous appeared only in the second half of the 20th century. The first digitally operated and programmable robot, the Unmated, was installed in 1961 to lift hot pieces of metal from a die casting machine and stack them. Commercial and industrial robots are widespread today and used to perform jobs more cheaply, more accurately and more reliably, than humans. They are also employed in some jobs which are too dirty, dangerous, or dull to be suitable for humans. Robots are widely used in manufacturing, assembly, packing and packaging, mining, transport, earth and space exploration, surgery, weaponry, laboratory research, safety, and the mass production.

2.4 APPLICATIONS

As more and more robots are designed for specific tasks this method of classification becomes more relevant. For example, many robots are designed for assembly work, which may not be readily adaptable for other applications. They are termed as "assembly robots". For seam welding, some suppliers provide complete welding systems with the robot i.e. the welding equipment along with other material handling facilities like turntables, etc. as an integrated unit. Such an integrated robotic system is called a "welding robot" even though its discrete manipulator unit could be adapted to a variety of tasks. Some robots are specifically designed for heavy load manipulation, and are labelled as "heavy-duty robots".

Current and potential applications are given below:

A)Military robots:

Military robots are autonomous robots or remote-controlled mobile robots designed for military applications, from transport to search & rescue and attack. Some such systems are currently in use, and many are under development.

B)Industrial robots:

Robots are increasingly used in manufacturing (since the 1960s). According to the Robotic Industries Association US data, in 2016 the automotive industry was the main customer of industrial robots with 52% of total sales. In the auto industry, they can amount for more than half of the "labor". There are even "lights off" factories such as an IBM keyboard manufacturing factory in Texas that was fully automated as early as 2003.

C)Cobots:

Cobots, or collaborative robots, are robots intended to interact with humans in a shared space or to work safely in close proximity. Cobots stand in contrast to traditional industrial robots which are designed to work autonomously with safety assured by isolation from human contact. Cobot safety may rely on lightweight construction materials, rounded edges, and limits on speed or force. Safety may also require sensors and software to assure good collaborative behavior.

D)Construction robots:

Construction robots can be separated into three types: traditional robots, robotic arm, and robotic exoskeleton.

E)Agricultural robots (Agrobionts):

The use of robots in agriculture is closely linked to the concept of AI-assisted precision agriculture and drone usage. 1996-1998 research also proved that robots can perform a herding task.

F)Kitchen automation:

Commercial examples of kitchen automation are Flippy (burgers), Zume Pizza (pizza), Cafe X (coffee), Makr Shakr (cocktails), Frobot (frozen yogurts) and Sally (salads). Home examples are Rotimatic (flatbreads baking) and Boris (dishwasher loading).

G)**Robot combat for sport:**

Hobby or sport event where two or more robots fight in an arena to disable each other. This has developed from a hobby in the 1990s to several tv series worldwide.

CHAPTER-3
INTRODUCTION TO PROJECT

3.1 MOTIVATION

Detecting, localizing or following targets is at the core of numerous robotic applications, in both adversarial and cooperative contexts. Much work has been devoted in various research communities to such problems and the related deci- sional issues, which are often referred to as "pursuit-evasion" problems in the literature. This very evocative term actually encompasses a variety of scenarios that pertain either to mono- or multi-robot contexts, considering either a single or multiple targets, and whose objective is either to detect, to capture or to track them. On the other hand, other similar problems are named differently and make use of specific vocabulary, *e.g.* surveillance, search or tracking. This is partly explained by the different application contexts considered (industrial, civilian or military), and by the fact that different communities tackled them with different standpoints (e.g. sensor data processing, symbolic or geometric task planning, task allocation, game theory, etc.)

The variety of target related robotics problems and proposed approaches has motivated a vast amount of contributions, and several surveys focused on specific problems are available. In this article, our objective is to go beyond the frontiers of specific communities and specific problems, and to enlarge the scope of prior surveys.

To prior knowledge on the target position, and yields the two main classes of problems that as shown in figure 3.1, that often occur in sequence: detecting targets on the one hand, and tracking detected targets on the other hand.

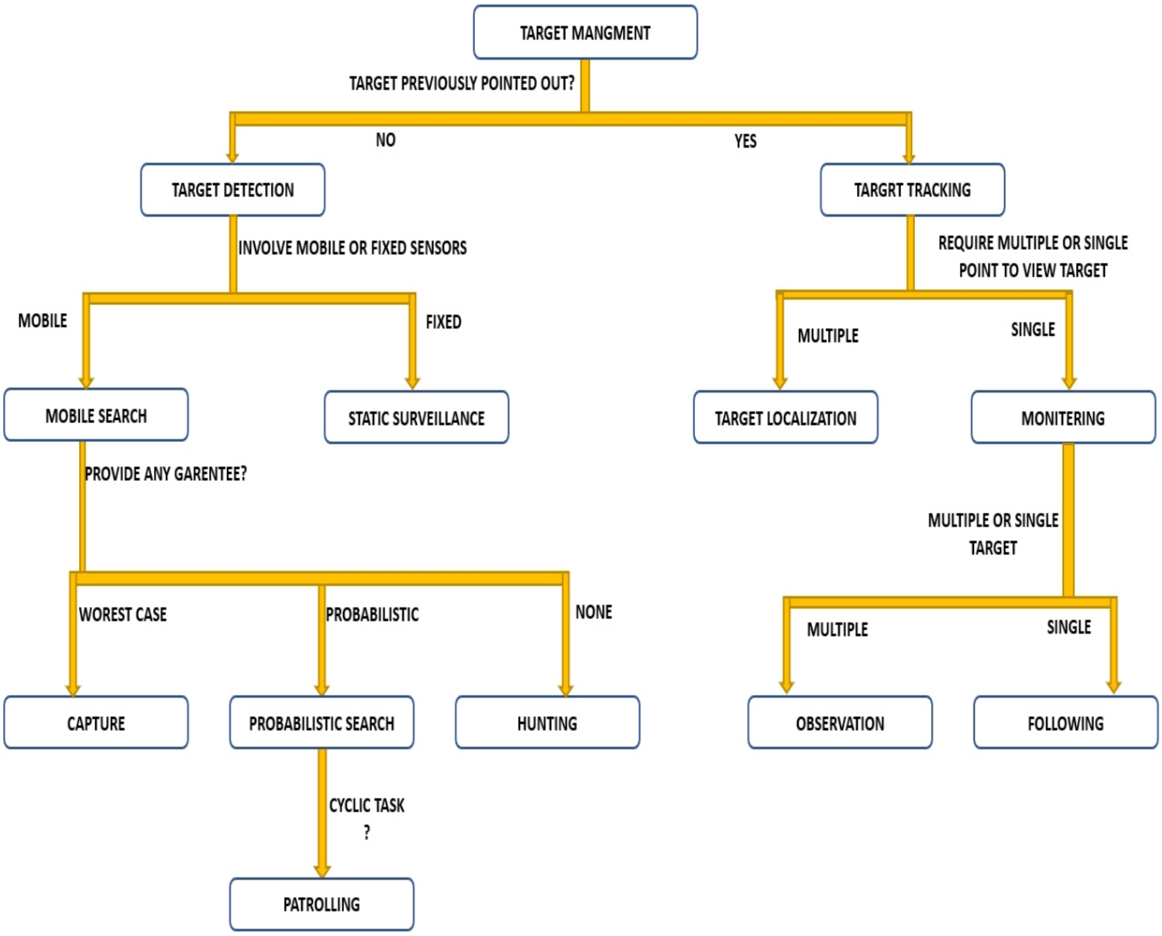


Figure3.1 Proposed taxonomy of the target management problems.

3.2 TARGET DETECTION

Target detection problems consist in finding (detecting) a target in a given environment. They may concern one or several targets and may be tackled with one or multiple sensors, either by actively sweeping the environment with mobile sensors, or by monitoring signals emitted from fixed static sensors.

We refer to this later class of problems as static surveillance: it mainly involves sensor positioning strategies, which often come to partition the environment and accordingly distribute sensors within this environment.

When mobile sensors are exploited, the problem is strongly related to path planning, and we refer to it as mobile search. Such problems can be addressed either locally or globally. Depending on the models, the assumptions and the approaches, some authors try to provide worst-case guarantees for the performance (capture), whereas some provide probabilistic guarantees (probabilistic search) and others do not provide any guarantee at all (hunting). The search may have a cyclic aspect (patrolling), although this presents no interest for capture or hunting. Note that for each of these problems, a variant that consists in "surrounding" the target in order to prevent its evasion is sometimes considered, instead of merely watching it, or catching it.

Static surveillance:

In static surveillance, the environment is necessary known and the objective is to optimally position a set of fixed sensors.

Capture:

In the capture problem, optimality and complete- ness are essential characteristics. The goal is to clear a given known area while providing a worst-case guarantee, meaning that if a target is inside the considered area, it will be found, no matter what. There is no prior knowledge or assumption on the target location, targets may even have "super abilities" (like infinite speed), and pursuers try to surround them. Capture is often referred to as a pursuit-evasion, but also as search and secure, or as the cops and robbers' game, mostly when the solution relies on graph clearing.

Probabilistic Search:

The main difference between capture and probabilistic search is the absence of worst-case guarantee in the latter, in which probabilities of detection are assessed. The reason is mainly a lack of resources (robots or time) to tackle the worst-case problem, but it can also be a compromise between efficiency and the probability of occurrence of particularly difficult situations. Probabilistic Search exploits probability distributions over the model of the environment (of the target presence, of the target visibility, etc.) Most authors try to provide bounds on the probability of detecting/catching the target. The target model may either be adversarial or not, the latter being usually easier to deal with because of its lower algorithmic complexity. The non- adversarial target model is widely used in search and rescue scenarios, for which emergency and time constraints usually prevent performing an exhaustive search and impose priorities which is well handled by probabilistic models.

Patrolling:

When the mobile search is cyclic, it is denoted by patrolling. Patrolling may be seen as a cyclic version of probabilistic search, as it involves analyses of statistical perform- mince over time, and especially the time elapsed between two visits to the same point.

Hunting:

There are finally some cases where no guarantee at all is provided for detection or capture of the target, which we refer to as hunting. The absence of guarantee comes from the lack of resources (robots, time) or information – in the absence of which no useful probability models for the target location can be exploited, for instance. Hunting is often considered within a multi-robot context; related work in presented area.

3.3 TARGET TRACKING

The second major class of problems, target tracking, core- spends to the tasks that arise when one or several targets have been detected or assigned – often following the success of target detection tasks. Note that the target tracking task itself is an estimation and control problem, not a planning problem and is thus not addressed here. In our survey, coping with a target may imply keeping it in sight, to provide information on it (mainly to localize it over time, but identifying it can also b e an objective), or to catch it. In all cases tracker robots need to stay “close” to the targets, the required distance being zero when it comes to catch the targets. Coping with a single target may require one or more robots, depending on the context. It is for instance preferable to have multiple vantage points on each target to refine their locations.

Target localization:

In target localization problems, the goal is to track a target with several robots in order to improve knowledge about the target, in particular the precision of its estimated position. It is most often a multi-robot problem, in which case solutions involve selecting different points of view to maximize the information gain. Of course, data fusion, cooperation, communication and multi-robot localization are issues to be considered. Several targets may be involved, and several observation points of view on each target are often required. Multi-robot target localization has naturally gained interest with the development of research on multi- robot systems.

Following:

In the original version, a single purser (the Lion) is chasing a single evader (the Man) with the same speed. Many different versions of the problem have been defined, with different speeds, environment models, visibility conditions, etc. Follow- in embraces all these variations, which are tracking problems involving only one pursuer and one evader.

Observation:

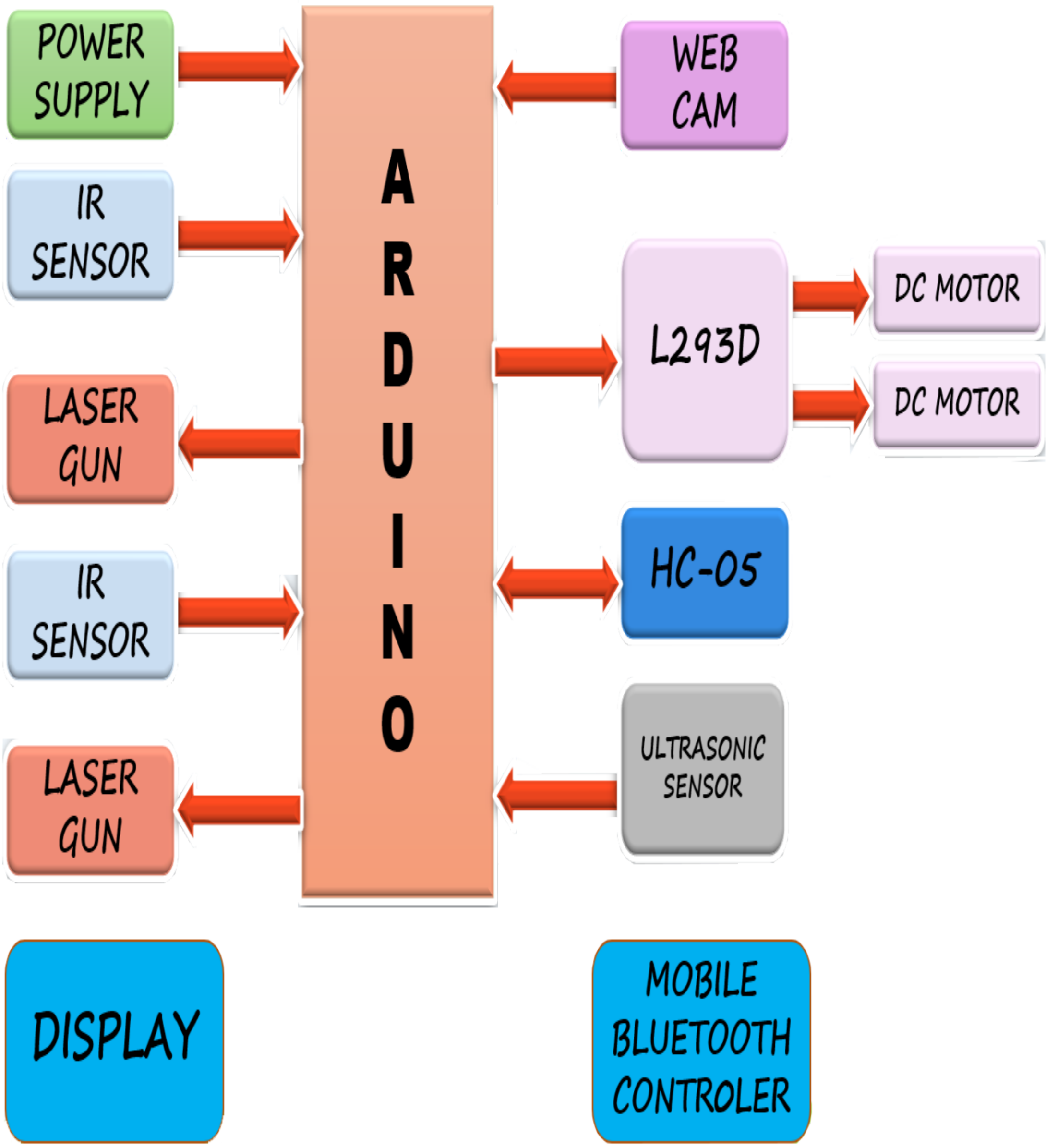
The observation problem is stated as follows: given several robots and several (moving) targets, how to control the robots in order to simultaneously observe all the targets, and if not possible, how to minimize the time during which any target is not observed by at least one of the robots.

Existing System:

On-board real robots: As stated in the introduction of this section, various algorithms have been integrated on-board robots. They show interesting performances, which advocates their maturity. For instance, Karla embeds a decentralized market-based approach on-board Pioneer robots (Karla et al, 2005), while Vieira runs algorithms on a team of iRobot Create robots, and a full network providing communication and "sensing-at-a-distance" abilities in the whole indoor end- virement (Vieira et al, 2009). Due to the poor sensors of the robots, Vieira uses wall-following motions, which is sufficient in the considered office environment. Katsilieris uses a large but simple outdoor environment (Katsilieris et al, 2010), whereas Durham's experiment takes place in an ad hoc indoor environment, using a previously tested architecture: the Multi-robot Integration Platform along with the Player-Stage framework (Durham et al, 2011).

Katsev et al. address the problem in relation to the sensing community (Katsev et al, 2011). They use a wall-following robot, without sufficient sensors that would allow precise mapping or localization. Introducing the notion of shadow (where an evader is not detected when behind an obstacle), they compute the global strategy on a map that allows area clearing, given that it is possible with a single robot in the considered environment. Their algorithms are implemented on the Lego NXT robot. The computed plans are worst-case guaranteed, yet the completeness of the algorithm has not been proven.

3.4 BLOCK DIAGRAM



3.2 block diagram of robot

3.5 TECHNICAL SPECIFICATIONS

3.5.1 HARDWARE

- Development Board : ARDUINO UNO
- Micro controller : ATMEGA328P, AT16U2
- Crystal Frequency : 16 MHz
- WEB CAM
- L293D
- DC MOTORS
- IR SENSORS
- UV SENSOR
- LEASER GUN
- HC-05

3.5.2 SOFTWARE

- ARDUINO IDE

3.6 WORKING

Robotic system for detecting and tracking moving objects. Such a system provides important capabilities for assistance of humans in various settings, e.g. home use, health care and transportation.

Bluetooth controlled robot is controlled by using Android mobile phone. Here we use button in android phone to control the robot in forward, backward, left and right directions. So here android phone is used as transmitting device and Bluetooth module placed on the kit is used as receiver. Android phone will transmit command using its in-built Bluetooth, so that robot can move in the required direction like moving forward, reverse, turning left, turning right and stop.in this system we placed a laser to target the enemies, objects etc. When left IR sensors that present in the robotic system activates when an obstacle come besides to it and indicates with a left laser fire. When obstacle comes besides to the right IR sensor that present in the robotic system will activates and displays as right

When an object that comes under below that the range that specified in the system that indicates that robot is stopped an obstacle will present will indicates.

Through ipweb cam we can see the objects that present Infront of the robot as live take decision what should be action that we should take. We can also take photos and video record for further investigation.

3.7 APPLICATIONS

Applications of robots:

1.Industrial

a) Logistics.
b) Manufacturing. (automobile)

2.Service

a) Medical (surgeries, patient care, research)
b) Home
c) Educational
d) Defense (border security, mission accomplishments)

3.Agriculture.

4.Mining, Excavation, Exploration

5.Survillence, Patrolling.

6.Rescue operations.

CHAPTER-4
MICROCONTROLLER

4.1 INTRODUCTION

Microcontroller as the name suggest, a small controller. They are like single chip computers that are often embedded into other systems to function as processing/controlling unit. For example, the control you are using probably has microcontrollers inside that do decoding and other controlling functions. They are also used in automobiles, washing machines, microwaves ovens, toys, etc… where automation is needed.

4.1.1 ARDUINO UNO MICROCONTROLLER

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means "One" in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5Vpin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows: -

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** the regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3.3V.**A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

MEMORY

The Atmega328 has 32 KB of flash memory for storing code (of which 0, 5 KB is used for the boot loader); it has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

INPUT AND OUTPUT

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

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt () function for details.

- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analog Write () function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED icon, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default, they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog Reference () function. Additionally, some pins have specialized functionality.

- **I2C: 4 (SDA) and 5 (SCL).** Support I2C (TWI) communication using the Wire library.

  There are a couple of other pins on the board:
- **AREF.** Reference voltage for the analog inputs. Used with analog Reference ().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

COMMUNICATION

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USBCOM drivers, and no external driver is needed. However, on Windows, an *.in file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

4.2 ARDUINO UNO BOARD

The Arduino Uno is a microcontroller board based on the ATmega328 as shown in figure 4.1. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

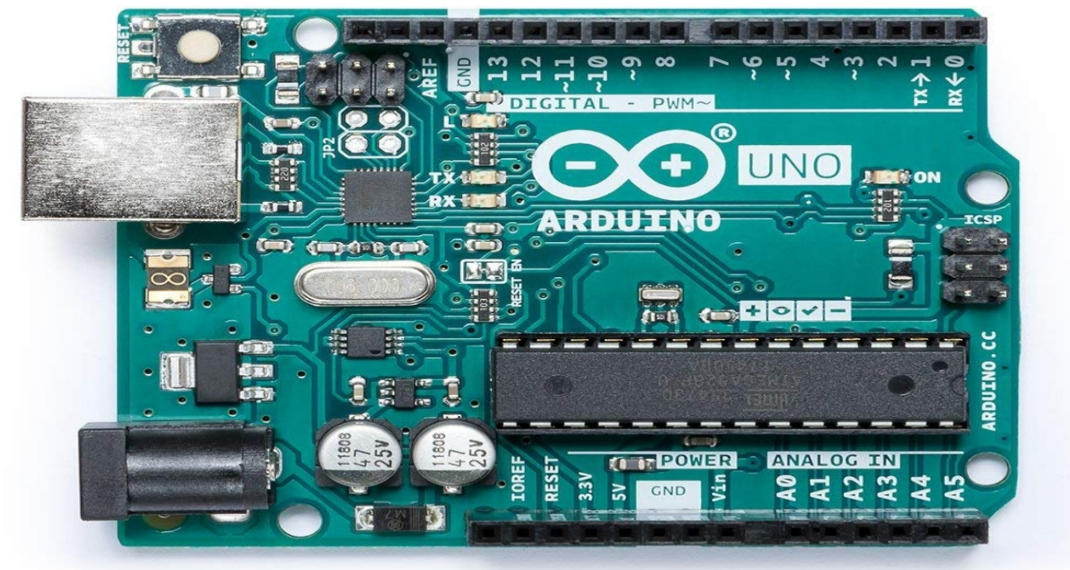


Figure.4.1 Arduino Uno board

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converters.

4.2.1. TECHNICAL SPECIFICATIONS

Microcontroller : ATmega328

Operating voltage : 3.3v

Input voltage : 7-12v

Input voltage (limits)	: 6-20v
Digital I/O pins	: 14(of which 6 provides PWM outputs
Analog input pins	: 6
DC current per I/O Pin	: 40mA
DC current for 3.3v pin	: 50mA
Flash memory	: 32KB (ATmega328) of which 0.5kb used by boot loader
SRAM	: 2KB (ATmega328)
EPROM	: 1KB (ATmega328)
Clock speed	: 16 MHz

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

USB Interface

Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection

External power supply

Arduino boards can be powered directly from the AC mains power supply by connecting it to the power supply (Barrel Jack)

Voltage Regulator

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

Crystal Oscillator

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz

Arduino Reset

It can reset your Arduino board, i.e., start your program from the beginning. It can reset the UNO board in two ways. First, by using the reset button on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET.

Pins:6-9 (3.3, 5, GND, VIN)

- 3.3V (6): Supply 3.3 output volt
- 5V (7): Supply 5 output volt
- Most of the components used with Arduino board works fine with 3.3 volt and 5 volts.
- GND (8) (Ground): There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- Vin (9): This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

Analog pins

The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

Main microcontroller

Each Arduino board has its own microcontroller . You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

The Atmega8U2 programmed as a USB-to-serial converter. "Uno" means "One" in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

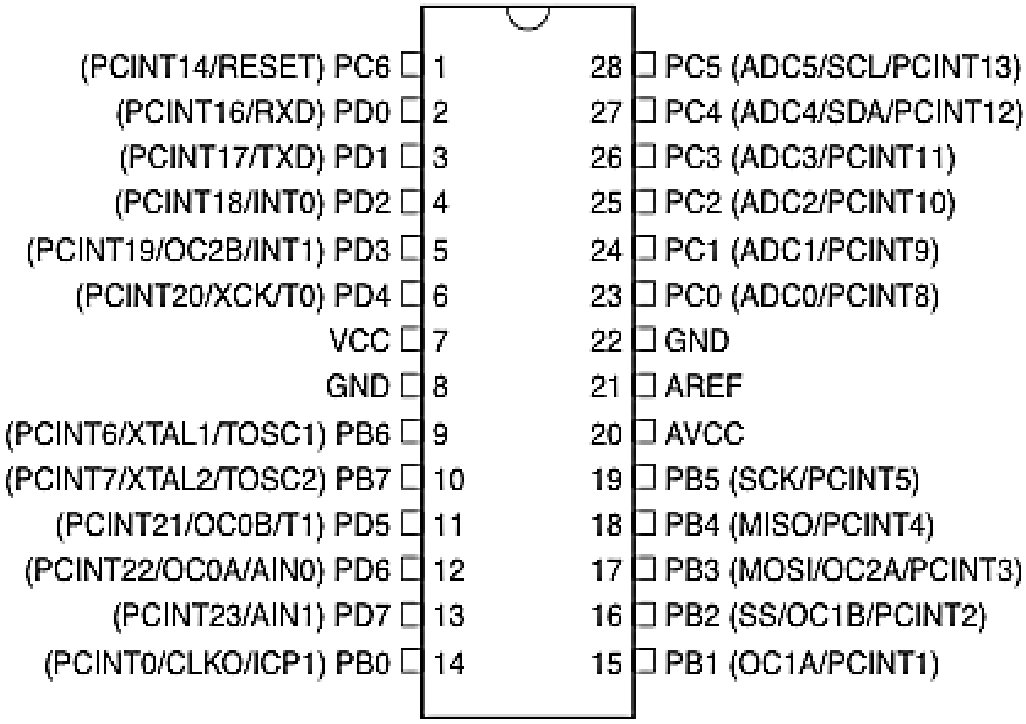


Figure.4.2 Pin diagram of ATmega microcontroller

4.2.2. PIN DESCRIPTION

VCC: Digital supply voltage.

GND: Ground.

Port B (PB [7:0]) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB [7:6] is used as TOSC [2:1] input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

Port C (PC [5:0])

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC [5:0] output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs,

Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a Reset.

Port D (PD [7:0])

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

AVCC

AVCC is the supply voltage pin for the A/D Converter, PC [3:0], and PE [3:2]. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter. Note that PC [6:4] use digital supply voltage, VCC.

AREF

AREF is the analog reference pin for the A/D Converter.

ADC [7:6] (TQFP and VFQFN Package Only)

In the TQFP and VFQFN package, ADC [7:6] serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

12. ICSP pin

Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

13. Power LED indicator

This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

14. TX and RX LEDs

On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

15. Digital I / O

The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.

16. AREF

AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins working.

CHAPTER-5

HARDWARE COMPONENTS

5.1 DC BATTERY

A battery is a device that can store electricity. Some are rechargeable, and some are not. They store direct current (DC) electricity.

A battery really means two or more wet or dry cells connected in series for more voltage, or in parallel for more current, although people often call a cell a battery. AA, AAA, C, and D batteries all have 1.5 volts. The voltage of a cell depends on the chemicals used while the amount of power or current it can supply also depends on how large the cell is; a bigger cell of a given type can supply more amps, or for a longer time.

The chemical reactions that occur in a battery are exothermic reactions and, thus, produce heat. For example, if you leave your laptop on for a long time, and then touch the battery, it will be warm or hot. However, the batteries used in laptops are called lithium-ion batteries and they sometimes do have a fire hazard (A few years ago, dell laptops that that were powered by lithium batteries began to catch fire, though this event was rare.).

Batteries come in lots of different shapes,sizes and voltages as shown in figure 5.1. It is possible, but not easy, to run wires to use an odd size battery for an odd purpose.

Batteries are always more costly/expensive than mains electricity. But mains electricity is not suitable for things that are mobile.

Bicycles have tail-lights that can be operated by batteries, and sometimes by a little generator powered by the wheels.

Hand and foot generators can be used to replace batteries in various devices, but they can be tire some.

Wind-up generators are now available to power small clockwork radios, clockwork torches, etc.

Since clockwork clocks have been around for hundreds of years, and batteries for two hundred, it is amazing that no-one thought of a clockwork torch until recently.

Rechargeable batteries are recharged by reversing the chemical reaction that occurs within the battery. But a rechargeable battery can only be recharged a given amount of time (recharge life). Even iPods, with built in batteries, cannot be recharged forever. Moreover, each time a battery is recharged, its ability to hold a charge is degraded a bit. Non-rechargeable batteries should not be charged as various caustic and corrosive substances can leak out, such as potassium hydroxide.

Figure 5.1 different shapes and sizes and voltages of Batteries

The very first batteries were invented in the middle east around 1000 B.C. Then they were buried and forgotten about. The first battery was invented in 1800 by Alessandro Volta. Nowadays, his battery is called the voltaic pile.

Later batteries were bottles with a fluid and some metal rods in them. People had to be careful not to turn these batteries upside-down so the fluid would spill.

In modern batteries, the fluid is "soaked up" in a kind of paste. And everything is put in a completely tight case: Because of this case, nothing can spill out of the battery. An exception is car batteries; they still have liquid inside.

Types of batteries

- Alkaline battery, "alkaline", not rechargeable
- Leclanche battery, "super heavy duty", not rechargeable
- Nickel metal-hydride battery, "NiMH", rechargeable
- Nickel cadmium battery, "NiCad", rechargeable
- Lead acid battery, rechargeable, car battery
- Lithium battery, unrechargeable, "coin cell"
- Lithium-ion battery, rechargeable, used in cell phones and laptops
- Mercury battery, unrechargeable
- Silver oxide battery, unrechargeable, watch battery

Alternatives to Batteries

- Solar cell

5.2 BLUETOOTH:

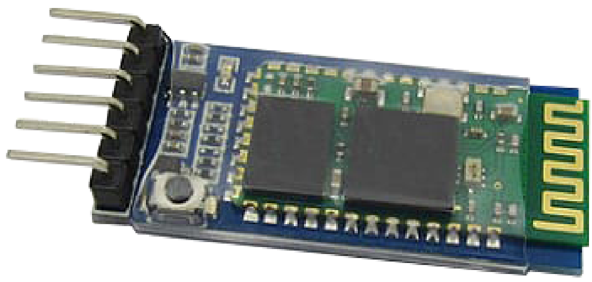


Figure 5.2 hc-05 Bluetooth module

This module enables you to wireless transmit & receive serial data. It is a drop-in replacement for wired serial connections allowing transparent two-way data communication. You can simply use it for serial port replacement to establish connection between MCU or embedded project and PC for data transfer. Bluetooth Core V2.0 compliant module with SPP.

The module is designed to be embedded in a host system which requires cable replacement function. Typically, the module could interface with a host through the UART port.

The module could be used in many different applications Examples:

- Hand held terminals
- Industrial devices
- Point-of-Sale systems
- PCs
- Personal Digital Assistants (PDAs)
- Computer Accessories
- Access Points
- Automotive Diagnostics Units

We supply module with 9600 baud rates in ready to use with PC. You will need a USB Bluetooth Adapter at PC side or Bluetooth Enabled Laptop to connect to our Bluetooth module. Module supplied by us with this setting: 9600 baud rate, Pair Code: 0000

The Bluetooth module works on 3.3V level only. High voltage like 5V will permanently damage the module, so please take care in using it.

If your application requires to be operated at 5V then use a LM1117-3.3 regulator to convert the 5V level to 3V3 level as required by module. Also protect the RXD pin against 5V TXD signal by inserting 1K resistor in series to module RXD pin.

If you wish to connect this module to PC's Serial port which is at RS232 level, then you need to add MAX232 circuit as shown above. Status LED flashes at different rates to indicate different status like searching, config, connected.

This module could both act a SPP master and a SPP slave. When in master mode, the module could search for all the working SPP slave devices around and the host could select which to connect. When it is in slave mode, it will listen for connection request from another SPP master device. Bluetooth UART provides the main interface to exchange data with other host system using the RS232 protocol. An external commands set is provided for the host system to control and configure AUBTM-20. Four signals are providing for UART function. TXD and RXD transmit data between AUBTM-20 and the host. NRTS and NCTS provides the RS232 hardware flow control mechanism. All UART pins are CMOS logic with signal levels of 0V to VDD. UART is initially configured to work at 9600 bps baud rate, 8-bit, no parity and 1 stop bit. The host could reconfigure the UART by issuing command

TABLE 5.1 BLUETOOTH SPECIFICATIONS

Pin name	Pin mode	Description
GND	GND	Ground
3V3	VDD	Power supply connection
PIO2	I/O	Programmable I/O lines
PIO3	I/O	Programmable I/O lines
NRTS	O	UART RTS (internal pull-up, active low)
RXD	I	UART RX (internal pull down)
PCMO	O	Synchronous 8 kbps data out (internal Pull down)
USB_D+	A	USB data plus (Internal 22-ohm serial resistor)
USB_D-	A	USB data minus (Internal 22-ohm serial resistor)
NCTS	I	UART CTS (internal pull down, active low)
PCMI	I	Synchronous 8 kbps data in (internal pull-down)
PCMC	I/O	Synchronous data clock (internal pull-down)
PCMS	I/O	Synchronous data strobe (internal pull-down)
GND	GND	Ground
GND	GND	Ground
3V3	VDD	Power supply connection
RES	I	Reset input (active low)
PIO6	I/O	Programmable I/O lines
PIO7	I/O	Programmable I/O lines
PIO4	I/O	Programmable I/O lines
NCSB	I	Chip selection for SPI (internal pull up, active low)
SCLK	I/O	SPI Clock (internal pull down)

MISO	O	SPI data output (pull down)
MOSI	I	SPI data input (pull down)
PIO5	I/O	Programmable I/O lines

5.3 IR SENSOR TECHNOLOGY

WHAT IS INFRARED?

Infrared is an energy radiation with a frequency below our eye's sensitivity, so we cannot see it. Even that we cannot "see" sound frequencies, we know that it exists, we can listen them. Infrared energy radiation scale is shown in figure 5.3.

Figure 5.3 infrared scale

Even that we cannot see or hear infrared, we can feel it at our skin temperature sensors.

When you approach your hand to fire or warm element, you will "feel" the heat, but you can't see it. You can see the fire because it emits other types of radiation, visible to your eyes, but it also emits lots of infrared that you can only feel in your skin.

5.3.1 IR GENERATION

To generate a 36 kHz pulsating infrared is quite easy, more difficult is to receive and identify this frequency. This is why some companies produce infrared receives, that contains the filters, decoding circuits and the output shaper, that delivers a square wave, meaning the existence or not of the 36kHz incoming pulsating infrared. pulsating 36kHz infrared circuit as shown in figure 5.4.

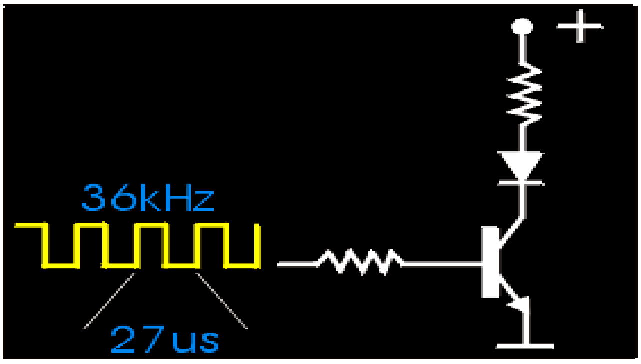


Figure 5.4 pulsating 36kHz infrared

It means that those 3 dollars small units, have an output pin that goes high (+5V) when there is a pulsating 36kHz infrared in front of it, and zero volts when there is not this radiation.

A square wave of approximately 27uS (microseconds) injected at the base of a transistor, can drive an infrared LED to transmit this pulsating light wave. Upon its presence, the commercial receiver will switch its output to high level (+5V). If you can turn on and off this frequency at the transmitter, your receiver's output will indicate when the transmitter is on or off as shown in figure 5.5.

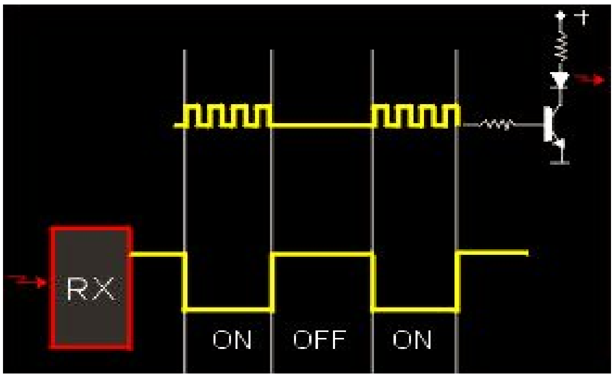


Figure 5.5 receiver's output

Those IR demodulators have inverted logic at its output, when a burst of IR is sensed it drives its output to low level, meaning logic level = 1.

The TV, VCR, and Audio equipment manufacturers for long use infra-red at their remote controls. To avoid a Philips remote control to change channels in a Panasonic TV, they use different codification at the infrared, even that all of them use basically the same transmitted frequency, from 36 to 50 kHz. So, all of them use a different combination of bits or how to code the transmitted data to avoid interference.

5.4 IR OBSTACLE SENSOR

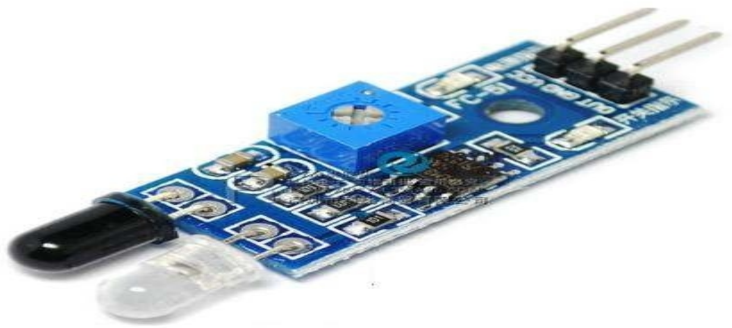


Figure 5.6 IR OBSTACLE SENSOR

The sensor module adaptable to ambient light, having a pair of infrared emitting and receiving tubes as shown in figure 5.6, transmitting tubes emit infrared certain frequency, when the direction of an obstacle is detected (reflection surface), the infrared reflected is received by the reception tube, After a comparator circuit processing, the green light is on, but the signal output interface output digital signal (a low-level signal), you can adjust the detection distance knob potentiometer, the effective distance range of 2 ~ 30cm, the working voltage of 3.3V- 5V. Detection range of the sensor can be obtained by adjusting potentiometer, with little interference, easy to assemble, easy to use features, can be widely used in robot obstacle avoidance, avoidance car, line count, and black and white line tracking and many other occasions. Ir obstacle sensor modules are as shown in figure 5.7.

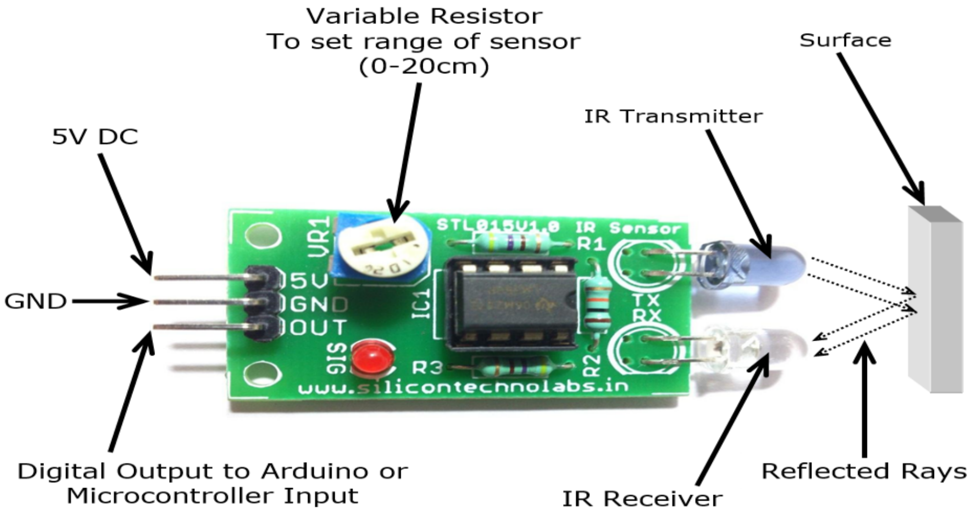


Figure 5.7 IR OBSTACLE SENSOR MODULES

Specification

1. When the module detects an obstacle in front of the signal, the green indicator lights on the board level, while the OUT port sustained low signal output, the module detects the distance 2 ~ 30cm, detection angle 35 °, the distance can detect potential is adjusted clockwise adjustment potentiometer, detects the distance increases; counter clockwise adjustment potentiometer, reducing detection distance.

2. The sensor active infrared reflection detection, target reflectivity and therefore

the shape is critical detection distance. Where the minimum detection distance black, white, maximum; small objects away from a small area, a large area from the Grand.

3. The sensor module output port OUT port can be directly connected to the microcontroller IO can also be directly drive a 5V relay; Connection: VCC-VCC; GND-GND; OUT-IO
4. Comparators LM393, stable;
5. The module can be 3-5V DC power supply. When the power is turned on, the red power indicator lights;
6. With the screw holes 3mm, easy fixed installation;
7. Board size: 3.2CM * 1.4CM
8. Each module has been shipped threshold comparator voltage adjusted by potentiometer good, non-special case, do not adjustable potentiometer.

Module Interface Description

1. VCC : 3.3V-5V external voltage (can be directly connected to 5v and 3.3v MCU)
2. GND : GND External
3. OUT : small board digital output interface (0 and 1)

5.5 ULTRASONIC OBSTACLE SENSOR:

HC-SR04 Ultrasonic Sensor

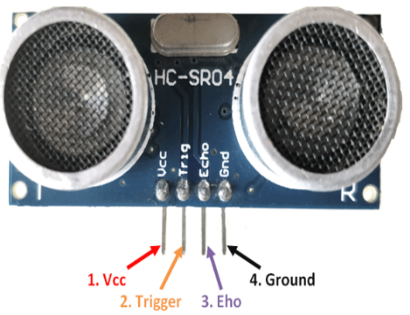


Figure 5.8 Ultrasonic Sensor HC SR04

TABLE 5.2:ULTRASONIC SENSOR PIN CONFIGURATION

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

Equivalent distance measuring Sensors

US transmitter Receiver pair, IR sensor module, IR sensor pair, IR Analog distance sensor,

HC-SR04 Ultrasonic Sensor - Working

As shown above figure 5.8 the HC-SR04 Ultrasonic (US) sensor is a 4-pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in figure 5.9.

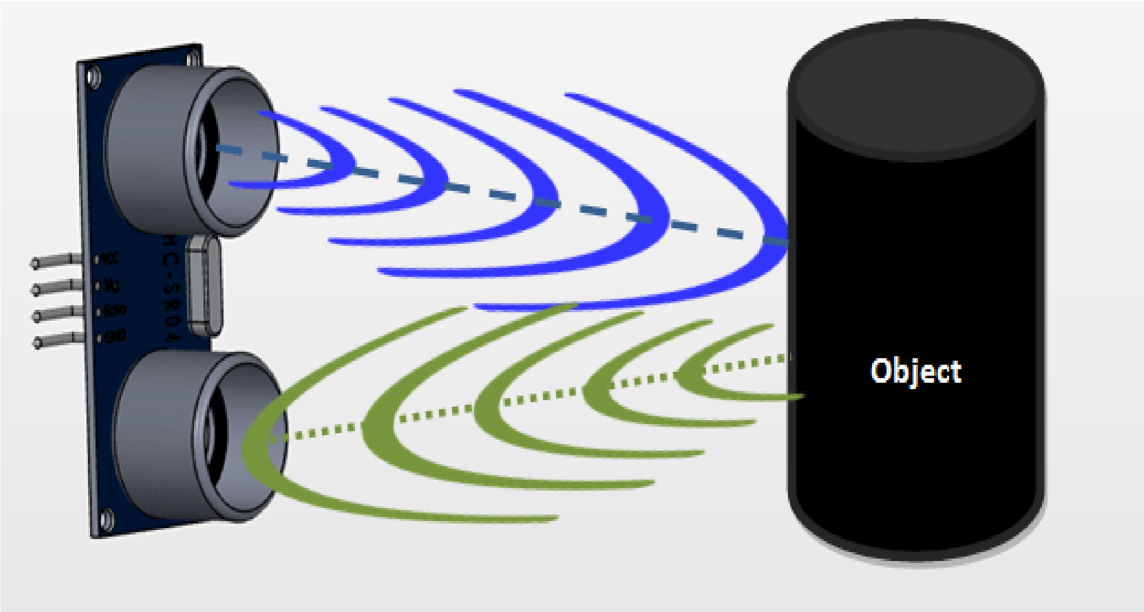


Figure 5.9 working diagram

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave, we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

How to use the HC-SR04 Ultrasonic Sensor

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information, the distance is measured as explained in the above heading.

Applications

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it
- Depth of certain places like wells, pits etc. can be measured since the waves can penetrate through water

5.6 L293D- Current Driver

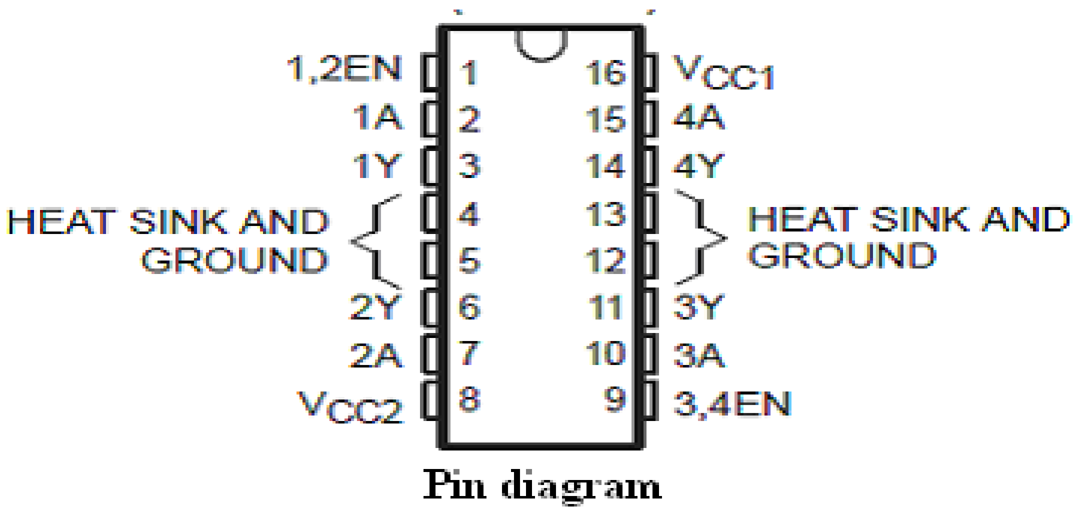


Figure 5.10 L293D drive

Features

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- Thermal Shutdown
- High-Noise-Immunity Inputs
- Functionally Similar to SGS L293 and SGS L293D
- Output Current 1 A Per Channel (600 mA for L293D)
- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

Description

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. pin diagram of L293D drive as shown in figure5.10. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo- Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications. On the L293, external high-speed output clamp diodes should be used for inductive transient suppression.

A VCC1 terminal, separate from VCC2, is provided for the logic inputs to minimize device power dissipation. The L293 and L293D are characterized for operation from 0 to 70 degree Celsius.

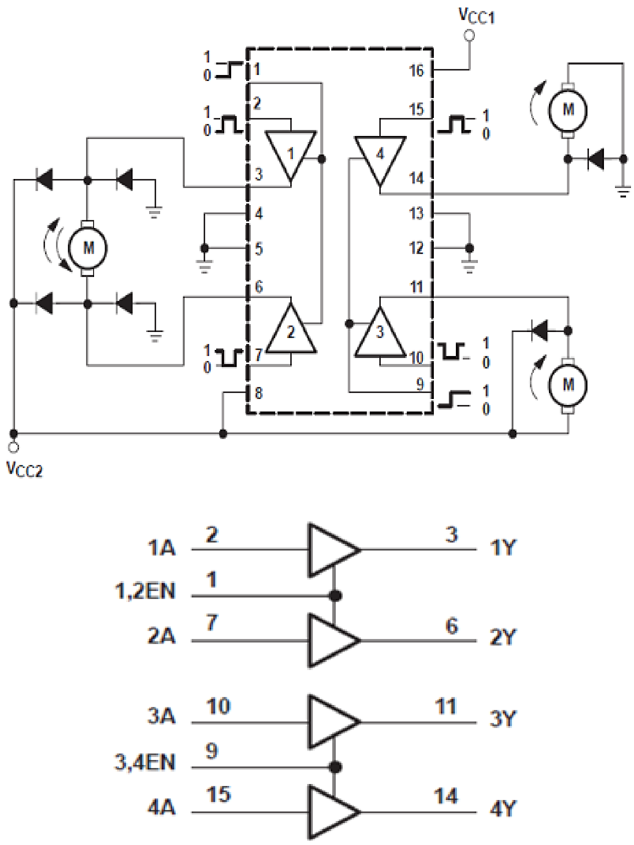


Figure 5.11 logic diagram

TABLE 5.3 FUNCTIONAL TABLE

INPUT		OUTPUT Y
A	EN	
H	H	H
L	H	L
X	L	Z

This chip contains 4 enable pins. Each enable pin corresponds to 2 inputs. Based on the input values given, the device connected to this IC works accordingly.

L293D Interfacing with Arduino:

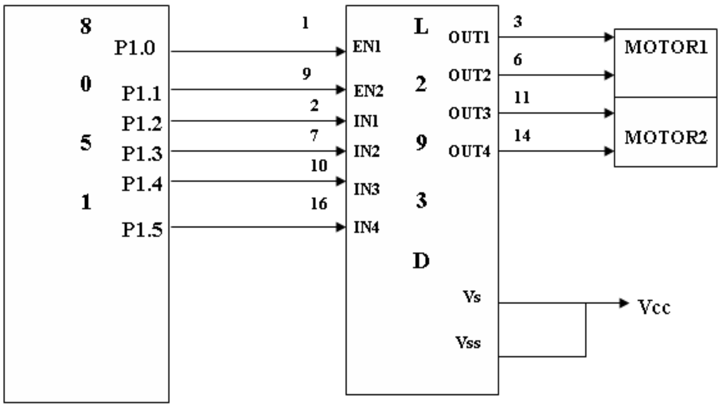


Figure 5.12 interfacing with Arduino

TABLE 5.4 CONTROLLING DIRECTIONS OF ROBOT

Controlling the Robot to obtain the different directions of movement

	Left Wheel	Right Wheel	Movement
1	Forward	Forward	Forward
2	Backward	Backward	Backward
3	Forward	Stop	Right Turn
4	Stop	Forward	Left Turn
5	Forward	Backward	Sharp Right Turn
6	Backward	Forward	Sharp Left Turn

The DC motor description is carried out in the next section. The L293D output pins will be connected to the two motors of Robot. Thus, the output of L293D depends on the input provided from the microcontroller and the enable pins. It should be remembered that unless the enable pins are not high, whatever input values given to L293D IC will not be applied to the motors in any way.

5.7 DC Motors:

Electric motors are used to efficiently convert electrical energy into mechanical energy. Magnetism is the basis of their principles of operation. They use permanent magnets, electromagnets, and exploit the magnetic properties of materials in order to create these amazing machines.

There are several types of electric motors available today. The following outline gives an overview of several popular ones. There are two main classes of motors: AC and DC. AC motors require an alternating current or voltage source (like the power coming out of the wall outlets in your house) to make them work. DC motors require a direct current or voltage source (like the voltage coming out of batteries) to make them work. Universal motors can work on either type of power. Not only is the construction of the motors different, but the means used to control the speed and torque created by each of these motors also varies, although the principles of power conversion are common to both.

Motors are used just about everywhere. In our house, there is a motor in the furnace for the blower, for the intake air, in the sump well, dehumidifier, in the kitchen in the exhaust hood above the stove, microwave fan, refrigerator compressor and cooling fan, can opener, garbage disposer, dish washer pump, clocks, computer fans, ceiling fans, and many more items.

In industry, motors are used to move, lift, rotate, accelerate, brake, lower and spin material in order to coat, paint, punch, plate, make or form steel, film, paper, tissue, aluminum, plastic and other raw materials.

They range in power ratings from less than 1/100 hp to over 100,000 hp. The rotate as slowly as 0.001 rpm to over 100,000 rpm. They range in physical size from as small as the head of a pin to the size of a locomotive engine.

What happens when a wire carrying current is within a magnetic field?

Figure 5.13 left hand rule

This is the Left-Hand Rule for motors as shown in figure 5.13. The first finger points in the direction of the magnetic field (first - field), which goes from the North pole to the South pole.

The second finger points in the direction of the current in the wire (second - current). The thumb then points in the direction the wire is thrust or pushed while in the magnetic field (thumb - torque or thrust).

So, when a wire carrying current is perpendicular to a magnetic field, a force is created on the wire causing it to move perpendicular to the field and direction of current. The greater the current in the wire, or the greater the magnetic field, the faster the wire moves because of the greater force created. If the current in the wire is parallel to the magnetic field, there will be no force on the wire.

DC Motors:

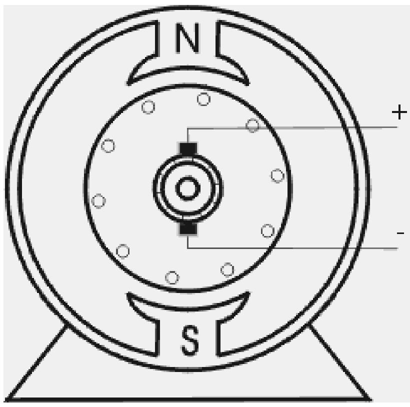


Figure 5.14 functional diagram motor

DC motors are fairly simple to understand. They are also simple to make and only require a battery or dc supply to make them run.

A simple motor has six parts, as shown in the diagram below:

- Armature or rotor
- Commutator
- Brushes
- Axle
- Field magnet
- DC power supply of some sort

An electric motor is all about magnets and magnetism: A motor uses magnets to create motion. If you have ever played with magnets you know about the fundamental law of all magnets: Opposites attract and likes repel. So, if you have two bar magnets with their ends marked "north" and "south," then the north end of one magnet will attract the south end of the other. On the other hand, the north end of one magnet will repel the north end of the other (and similarly, south will repel south). Inside an electric motor, these attracting and repelling forces create rotational motion**.**

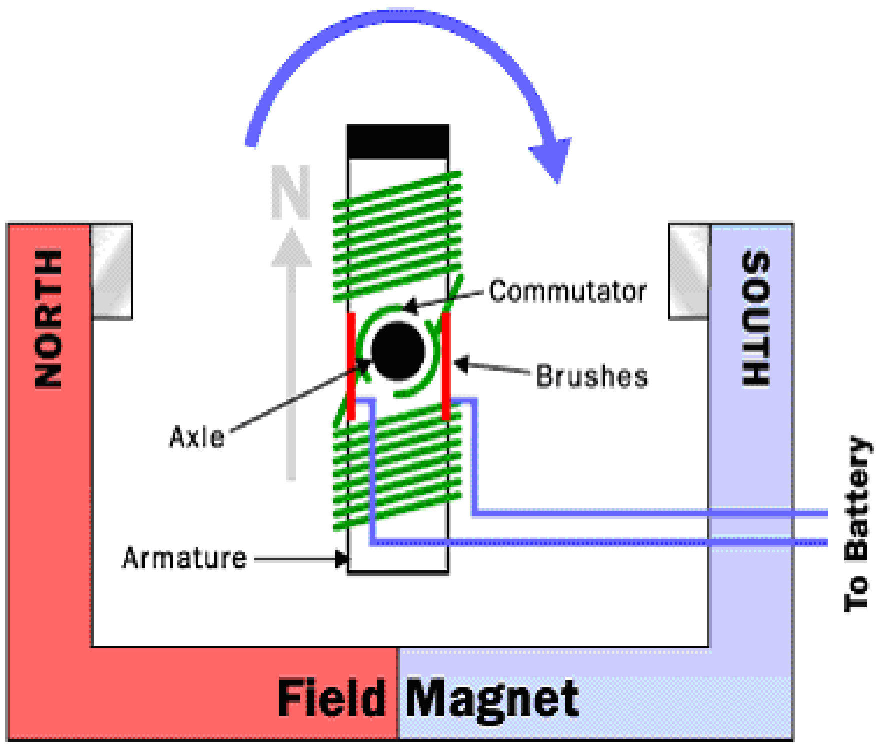


Figure 5.15 dc moto field diagram in different view

In figure 5.15 The armature (or rotor) is an electromagnet, while the field magnet is a permanent magnet (the field magnet could be an electromagnet as well, but in most small motors it is not in order to save power).

When you put all of these parts together, here is a complete electric motor:

In figure 5.16, the armature winding has been left out so that it is easier to see the commutator in action. The key thing to notice is that as the armature passes through the horizontal position, the poles of the electromagnet flip. Because of the flip, the north pole of the electromagnet is always above the axle so it can repel the field magnet's north pole and attract the field magnet's south pole.

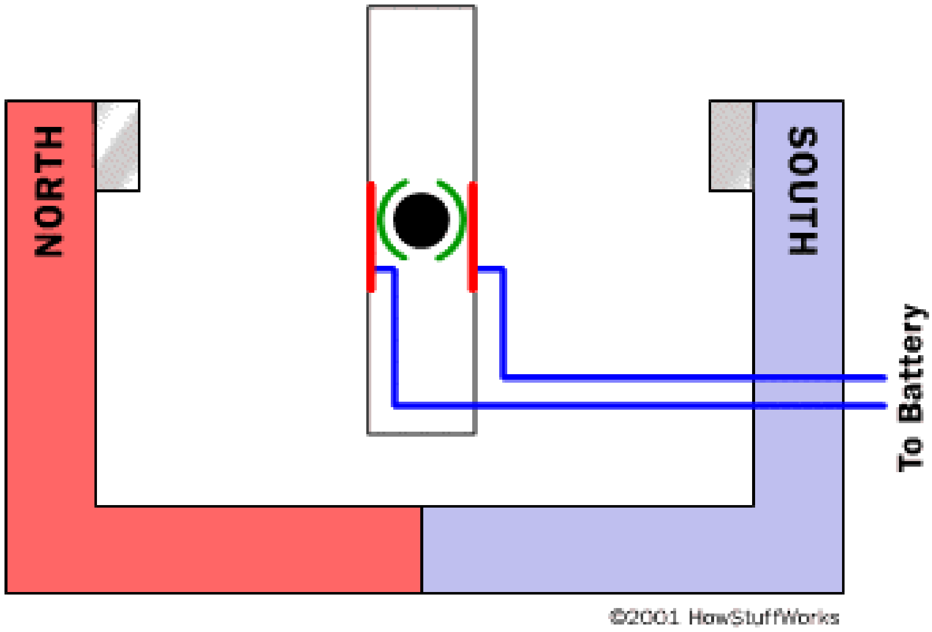


Figure 5.16 dc moto field diagram parts together

Even a small electric motor contains the same pieces described above: two small permanent magnets, a commutator, two brushes, and an electromagnet made by winding wire around a piece of metal. Almost always, however, the rotor will have three poles rather than the two poles as shown in this article. There are two good reasons for a motor to have three poles:

- It causes the motor to have better dynamics. In a two-pole motor, if the electromagnet is at the balance point, perfectly horizontal between the two poles of the field magnet when the motor starts, one can imagine the armature getting "stuck" there. This never happens in a three-pole motor.
- Each time the commutator hits the point where it flips the field in a two-pole motor, the commutator shorts out the battery (directly connects the positive and negative terminals) for a moment. This shorting waste energy and drains the battery needlessly. A three-pole motor solves this problem as well.

It is possible to have any number of poles, depending on the size of the motor and the specific application it is being used in.

5.7.1 Types of Motors

1.Split Phase

The split phase motor is mostly used for "medium starting" applications. It has start and run windings, both are energized when the motor is started. When the motor reaches about 75% of its rated full load speed, the starting winding is disconnected by an automatic switch.

Uses: This motor is used where stops and starts are somewhat frequent. Common applications of split phase motors include: fans, blowers, office machines and tools such as small saws or drill presses where the load is applied after the motor has obtained its operating speed.

2.Capacitor Start

This motor has a capacitor in series with a starting winding and provides more than double the starting torque with one third less starting current than the split phase motor. Because of this improved starting ability, the capacitor start motor is used for loads which are hard to start. It has good efficiency and requires starting currents of approximately five times full load current. The capacitor and starting windings are disconnected from the circuit by an automatic switch when the motor reaches about 75% of its rated full load speed.

Uses: Common uses include: compressors, pumps, machine tools, air conditioners, conveyors, blowers, fans and other hard to start applications.

5.7.2. Horsepower & RPM

Horsepower

Electric motors are rated by horsepower, the home shop will probably utilize motors from 1/4 HP for small tools and up to 5 HP on air compressors. Not all motors are rated the same, some are rated under load, others as peak horsepower and hence we have 5 HP compressors with huge motors and 5 Hp shoepacs with tiny little motors. Unfortunately, all 5 HP compressor motors are not equal in actual power either, to judge the true horsepower the easiest way is to look at the amperage of the motor. Electric motors are not efficient, most have a rating of about 50% due to factors such as heat and friction and some may be as high as 70%.

This chart will give a basic idea of the true horse power rating compared to the ampere rating. Motors with a higher efficiency rating will draw fewer amps, for example a 5 HP motor with a 50% efficiency rating will draw about 32 amps at 230 VACS compared to about 23 amps for a motor with a 70% rating.

TABLE 5.5 HORSE POWER RATING COMPARED TO THE AMPERE RATING

TRUE HP	AMPS at 115VAC	AMPS at 230 VAC
¼	3.2-2.3	1.6-1.2
1/3	4.3-3.1	2.2-1.5
½	6.5-4.6	3.2-2.3
¾	9.7-7.0	4.9-3.5
1	13.0-9.3	6.5-4.6
1 ½	19.5-13.9	9.7-7.0
2	25.9-18.5	13.0-9.3
5	64.9-46.3	32.4-23.2

A quick general calculation when looking at a motor is 1 HP = 10 amps on 110 volts and 1 HP = 5 amps on 220 volts.

RPM

The shaft on a typical shop motor will rotate at either 1725 or 3450 RPM (revolutions per minute).

The speed of the driven machine will be determined by the size of pulleys used, for example a 3450 RPM motor can be replaced by a 1750 RPM motor if the diameter of the pulley on the motor is doubled. The opposite is true as well but if the pulley on the 1750 RPM motor is small it is not always possible to replace it with one half the size. It may be possible to double the pulley size on the driven machine if it uses a standard type of pulley, (not easily done on air compressors for example).

Electronic speed reducers such as the ones sold for routers will not work on induction type motors.

Phase, Voltage & Rotation

Whether or not you can use a motor will likely depend on these factors.

Single Phase

Ordinary household wiring is single phase, alternating current. Each cycle peaks and dips as shown. To run a three-phase motor a phase converter must be used, usually this is not practical, it is often less expensive to change the motor on a machine to a single-phase style.

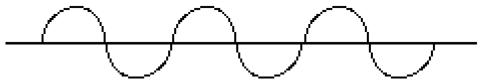


Figure 5.17 Single Phase

Three Phase

This is used in industrial shops, rather than peaks and valleys the current supply is more even because of the other two cycles each offset by 120 degrees.

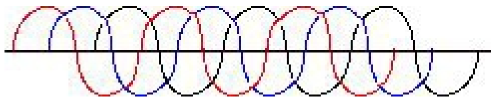


Figure 5.18 Three Phase

Voltage

Many motors are dual voltage i.e., by simply changing the wiring configuration, they can be run on 110 volts or 220 volts. Motors usually run better on 220 volts, especially if there is any line loss because of having to use a long wire to reach the power supply.

Motors are available for both AC and DC current, our typical home wiring will be AC. There are DC converters available which are used in applications where the speed of the motor is controlled.

Rotation

The direction the shaft rotates can be changed on most motors by switching the right wires. The direction of rotation is usually determined by viewing the motor from the shaft end and is designated as CW (clockwise) or CCW (counter-clockwise).

Inside the Wipers

The wipers combine two mechanical technologies to perform their task

1. A combination electric motor and worm gear reduction provides power to the wipers.
2. A neat linkage converts the rotational output of the motor into the back-and-forth motion of the wipers.

On any gear, the ratio is determined by the distances from the center of the gear to the point of contact. For instance, in a device with two gears, if one gear is twice the diameter of the other, the ratio would be 2:1.

One of the most primitive types of gears we could look at would be a wheel with wooden pegs sticking out of it.

The problem with this type of gear is that the distance from the center of each gear to the point of contact changes as the gears rotate. This means that the gear ratio changes as the gear turns, meaning that the output speed also changes. If you used a gear like this in your car, it would be impossible to maintain a constant speed you would be accelerating and decelerating constantly.

Worm gears

Figure 5.19 worm gears

These are used when large gear reductions are needed. It is common for worm gears to have reductions of 20:1, and even up to 300:1 or greater.

Many worm gears have an interesting property that no other gear set has: the worm can easily turn the gear, but the gear cannot turn the worm. This is because the angle on the worm is so shallow that when the gear tries to spin it, the friction between the gear and the worm holds the worm in place.

Motor and Gear Reduction

It takes a lot of force to accelerate the wiper blades back and forth across the windshield so quickly. In order to generate this type of force, a worm gear is used on the output of a small electric motor.

The worm gear reduction can multiply the torque of the motor by about 50 times, while slowing the output speed of the electric motor by 50 times as well. The output of the gear reduction operates a linkage that moves the wipers back and forth.

Inside the motor/gear assembly is an electronic circuit that senses when the wipers are in their down position. The circuit maintains power to the wipers until they are parked at the bottom of the windshield, and then cuts the power to the motor. This circuit also parks the wipers between wipes when they are on their intermittent setting.

Linkage

A short cam is attached to the output shaft of the gear reduction. This cam spins around as the wiper motor turns. The cam is connected to a long rod; as the cam spins, it moves the rod back and forth. The long rod is connected to a short rod that actuates the wiper blade on the driver's side. Another long rod transmits the force from the driver-side to the passenger-side wiper blade.

TABLE 5.6 OPERATIONAL SPECIFICATIONS

S.No	Feature	wiper motor
1	Voltage	12V
2	Torque	36Nm
3	Current	4A
4	Power	17HP
5	Speed	80RPM
6	Angle of wipe	100 deg

Description of the wiper motors selected

The motor is two pole design having high energy permanent magnets, together with a gear box housing, having two stages of gear reduction. power from the motor is a transferred by a three start, on an extension of the armature shaft through a two-stage gear system.

A ball bearing system is provided on the commutator end of the armature to minimize the friction losses and thereby increase torque of the wiper motor. Power from the final gear arm spindles. A special inbuilt limit switch ensures in applying regenerative braking to the OFF position.

Thermal protector is connected in series with armature to avoid burning of armature under locked position. Consistent parking of the wiper arms and blades in the correct position is there by ensured. The side on which the arms come to rest is preset to requirements.

Electrical connections are made to the motor via a non-reversible in line plug and socket assembly. This type of connections ensures that the correct motor polarity is maintained when the motor is connected to the vehicle wiring. The wiper motor incorporates radio interference capacitor.

5.8 LASER DIODE (650NM)

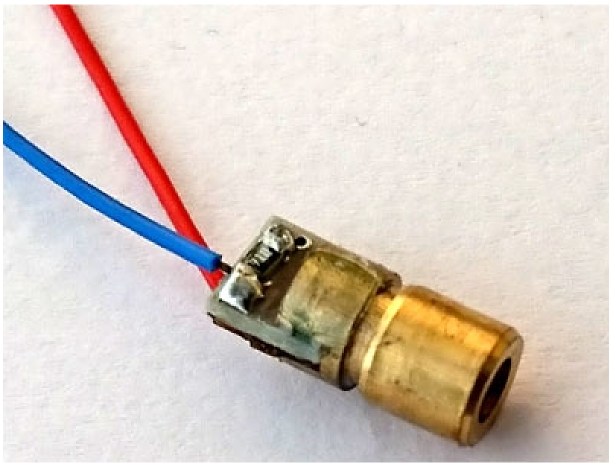


Figure 5.20 Laser Diode (650nm)

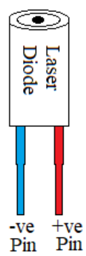


Figure 5.21 2 pin laser diode pinout

Features

- RoHS (Restriction of Hazardous Substances) Compliant

- Quality level is high
- Cost is economical
- Wavelength from 635 nm to 660 nm
- Rise and fall time is 0.5ns
- Package available: - TO-18(dia. 5.6mm), TO-5 (dia. 9 mm)

Technical Specification

- Operating Temperature: - -10～+40 ℃
- Storage Temperature: - -15～+85 ℃
- Output power (Po): - 5 mW
- Normal and maximum operating voltage is 2.2 and 2.7 respectively
- Threshold current in minimum, normal and maximum condition are 15, 20 and 30mA
- Operating current is 65 to 80mA
- Beam Angle deviation: -
- For both parallel and perpendicular condition its between -3 to 3 degrees
- Beam divergence: -
    - For parallel condition it’s between 8 to 12 degree
    - For perpendicular condition it’s between 23 to 32 degree

5.8.1 CHARACTERISTICS CURVES

Temperature Effect on Operation of Laser Diode

This graph is between Optical output power v/s forward current. It’s clear from the graph that laser output will only be visible if obtained above the threshold value of the laser diode. Before the threshold value the output of the laser diode is zero. After the threshold value the output of laser diode increase with slightly increase in forward voltage. The effect of temperature in the operation of Laser Diode is shown in graph below:

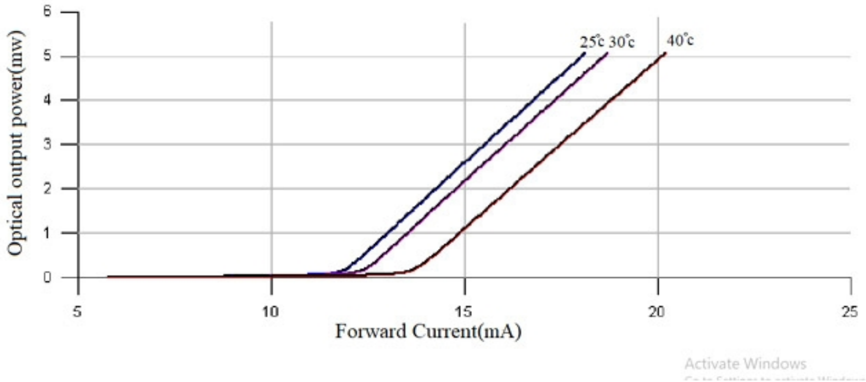


Figure 5.22 effect of temperature

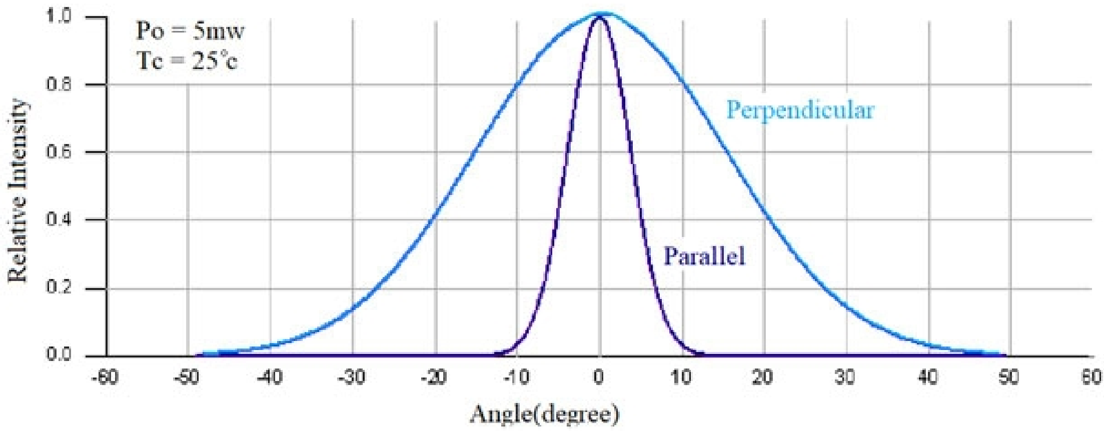


Figure 5.23 Laser Beam Divergence in parallel and perpendicular plane

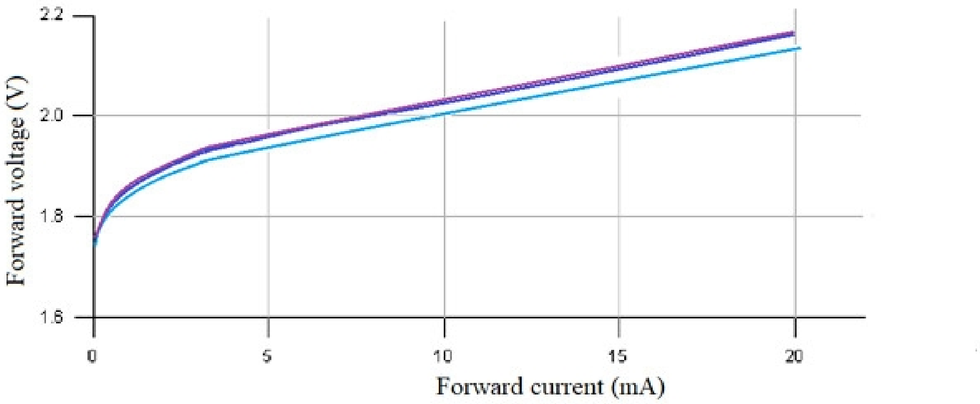


Figure 5.24 Forward voltage v/s Forward current

If the direction of moving current is forward and the output is continuously increasing, after the kink level the laser face a sudden breakdown which is the COD (Catastrophic Optical Damage) level. At this level due to high optical density the crystal at the face of diode melts. At the time of manufacturing of Red lasers, a special care is taken to avoid surge like static electricity and increase in current, because in red laser the oscillation is occur with the low power of 2 to 3 mW even after the breakdown. As the element is damaged the laser gets damaged or not able to work.

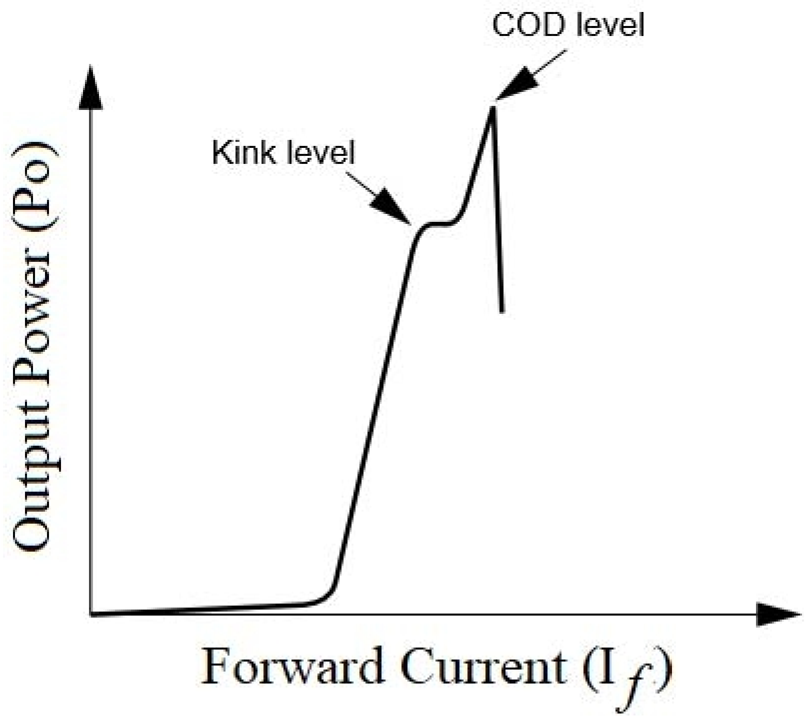


Figure 5.25 Output Power v/s Forward Current

How to use a Laser Diode?

If we want to operate a Laser diode then we must have laser diode driver circuit. As it helps in limiting current then supply it to laser diode. A laser diode can only work properly with the help of this circuit, if we directly connect it to the supply, because of having more current it will damage and if the value of current is low then the laser diode will not operate. Laser Diode driver circuit helps in providing a correct value of current to operate the laser diode.

The first capacitor in the circuit filter the High-frequency noise from the DC supply. The second capacitor works as power load balancer used to filter the fluctuating signals of output voltage. And a voltage regulator IC is used to provide a fix output voltage and we can adjust the output voltage by changing the value of resistor. You can use a potentiometer instead of resistor (R2) to adjust the intensity of laser light.

Applications

- Industrial applications: Engraving, cutting, scribing, drilling, welding, etc.
- Medical applications: to remove unwanted tissues, diagnostics of cancer cells using fluorescence, dental medication.
- Telecommunication
- Military application
- Data storage

CHAPTER-6

ARDUINO IDE

6.1 INTRODUCTION

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

6.1.1. ARDUINO DATA TYPES

Data types in C refers to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in the storage and how the bit pattern stored is interpreted.

The following table provides all the data types that you will use during Arduino programming.

Void: The void keyword is used only in function declarations. It indicates that the function is expected to return no information to the function from which it was called.

Example:

```
Void Loop ()
{
// rest of the code
}
```

Boolean: A Boolean holds one of two values, true or false. Each Boolean variable occupies one byte of memory.

Example:

Boolean state= false; // declaration of variable with type Boolean and initialize it with false.

Boolean state = true; // declaration of variable with type Boolean and initialize it with false.

Char: A data type that takes up one byte of memory that stores a character value. Character literals are written in single quotes like this: 'A' and for multiple characters, strings use double quotes: "ABC".

However, characters are stored as numbers. You can see the specific encoding in the ASCII chart. This means that it is possible to do arithmetic operations on characters, in which the ASCII value of the character is used. For example, 'A' + 1 has the value 66, since the ASCII value of the capital letter A is 65.

Example:

Char chr_a = ‘a’; //declaration of variable with type char and initialize it with character a.

Char chr_c = 97; //declaration of variable with type char and initialize it with character 97

Unsigned char: Unsigned char is an unsigned data type that occupies one byte of memory. The unsigned char data type encodes numbers from 0 to 255.

Example:

Unsigned Char chr_y = 121; // declaration of variable with type Unsigned char and initialize it with character y

Byte: A byte stores an 8-bit unsigned number, from 0 to 255.

Example:

Byte m = 25; //declaration of variable with type byte and initialize it with 25

Int: Integers are the primary data-type for number storage. **Int** stores a 16-bit (2-byte) value. This yields a range of -32,768 to 32,767 (minimum value of -2^15 and a maximum value of (2^15) - 1).

The **int** size varies from board to board. On the Arduino Due, for example, an **int** stores a 32-bit (4-byte) value. This yields a range of -2,147,483,648 to 2,147,483,647 (minimum value of -2^31 and a maximum value of (2^31) - 1).

Example:

Int counter = 32; // declaration of variable with type int and initialize it with 32.

Unsigned int: Unsigned ints (unsigned integers) are the same as int in the way that they store a 2-byte value. Instead of storing negative numbers, however, they only store positive values, yielding a useful range of 0 to 65,535 (2^16) - 1). The Due stores a 4-byte (32-bit) value, ranging from 0 to 4,294,967,295 (2^32 - 1).

Example:

Unsigned int counter= 60; // declaration of variable with type unsigned int and initialize it with 60.

Word:

On the Uno and other ATMEGA based boards, a word stores a 16-bit unsigned number. On the Due and Zero, it stores a 32-bit unsigned number.

Example

Word w = 1000; //declaration of variable with type word and initialize it with 1000.

Long: Long variables are extended size variables for number storage, and store 32 bits (4 bytes), from 2,147,483,648 to 2,147,483,647.

Example:

Long velocity= 102346; //declaration of variable with type Long and initialize it with 102346

Unsigned long: Unsigned long variables are extended size variables for number storage and store 32 bits (4 bytes). Unlike standard longs, unsigned longs will not store negative numbers, making their range from 0 to 4,294,967,295 (2^32 - 1).

Example:

Unsigned Long velocity = 101006; // declaration of variable with type Unsigned Long and initialize it with 101006.

Short:

A short is a 16-bit data-type. On all Arduinos (AT Mega and ARM based), a short store a 16-bit (2-byte) value. This yields a range of -32,768 to 32,767 (minimum value of -2^15 and a maximum value of (2^15) - 1).

Example:

Short Val= 13; //declaration of variable with type short and initialize it with 13

Float: Data type for floating-point number is a number that has a decimal point. Floating-point numbers are often used to approximate the analog and continuous values because they have greater resolution than integers. Floating-point numbers can be as large as 3.4028235E+38 and as low as 3.4028235E+38. They are stored as 32 bits (4 bytes) of information.

Example:

float num = 1.352;//declaration of variable with type float and initialize it with 1.352.

Double: On the Uno and other ATMEGA based boards, Double precision floating-point number occupies four bytes. That is, the double implementation is exactly the same as the float, with no gain in precision. On the Arduino Due, doubles have 8-byte (64 bit) precision.

Example:

double num = 45.352; // declaration of variable with type double and initialize it with 45.352.

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1:

First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image 6.1.

Figure.6.1 USB Cable

Step 2: Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

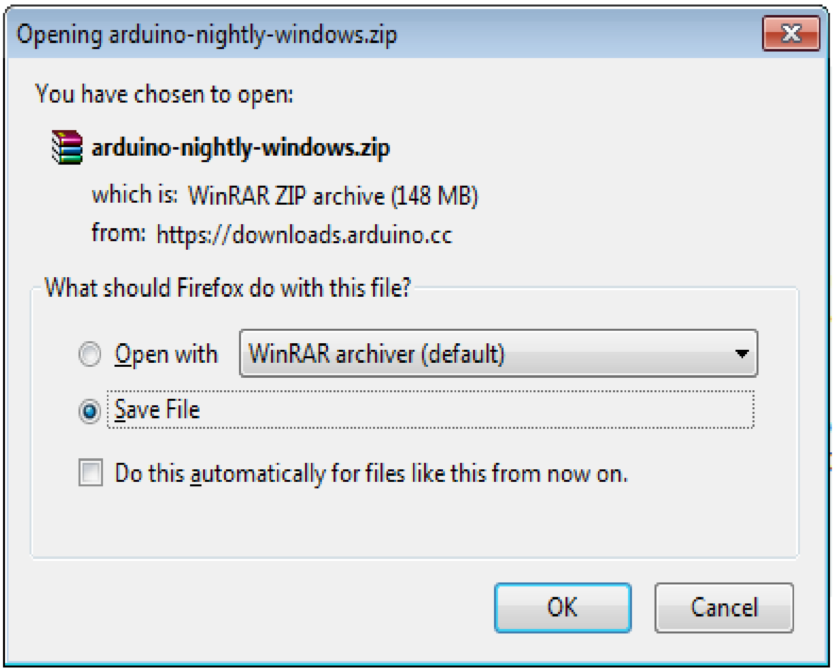


Figure.6.2 Downloading and saving the Arduino IDE

Step 3: Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4: Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). DoubleClick the icon to start the IDE.

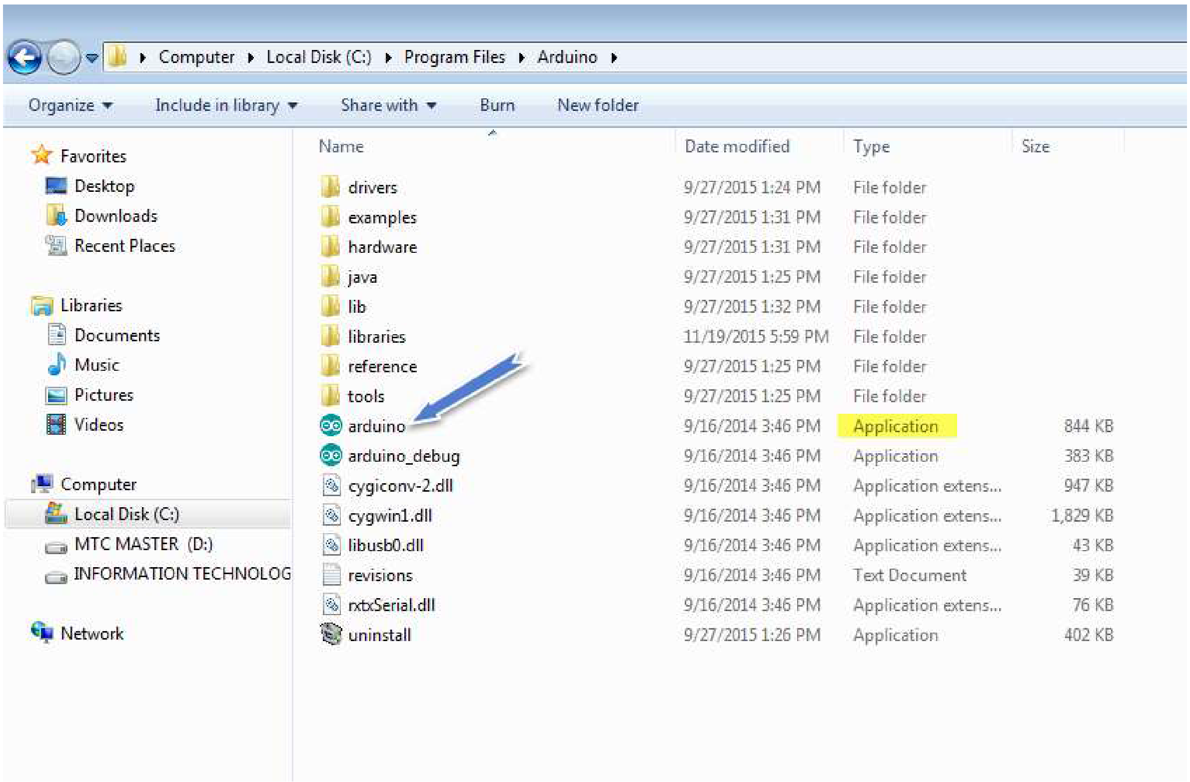


Figure.6.3 Launching of Arduino IDE

Step 5: Open your first project.

Once the software starts, you have two options:

- Create a new project.
- Open an existing project example.

To create a new project, select File --> New. To open

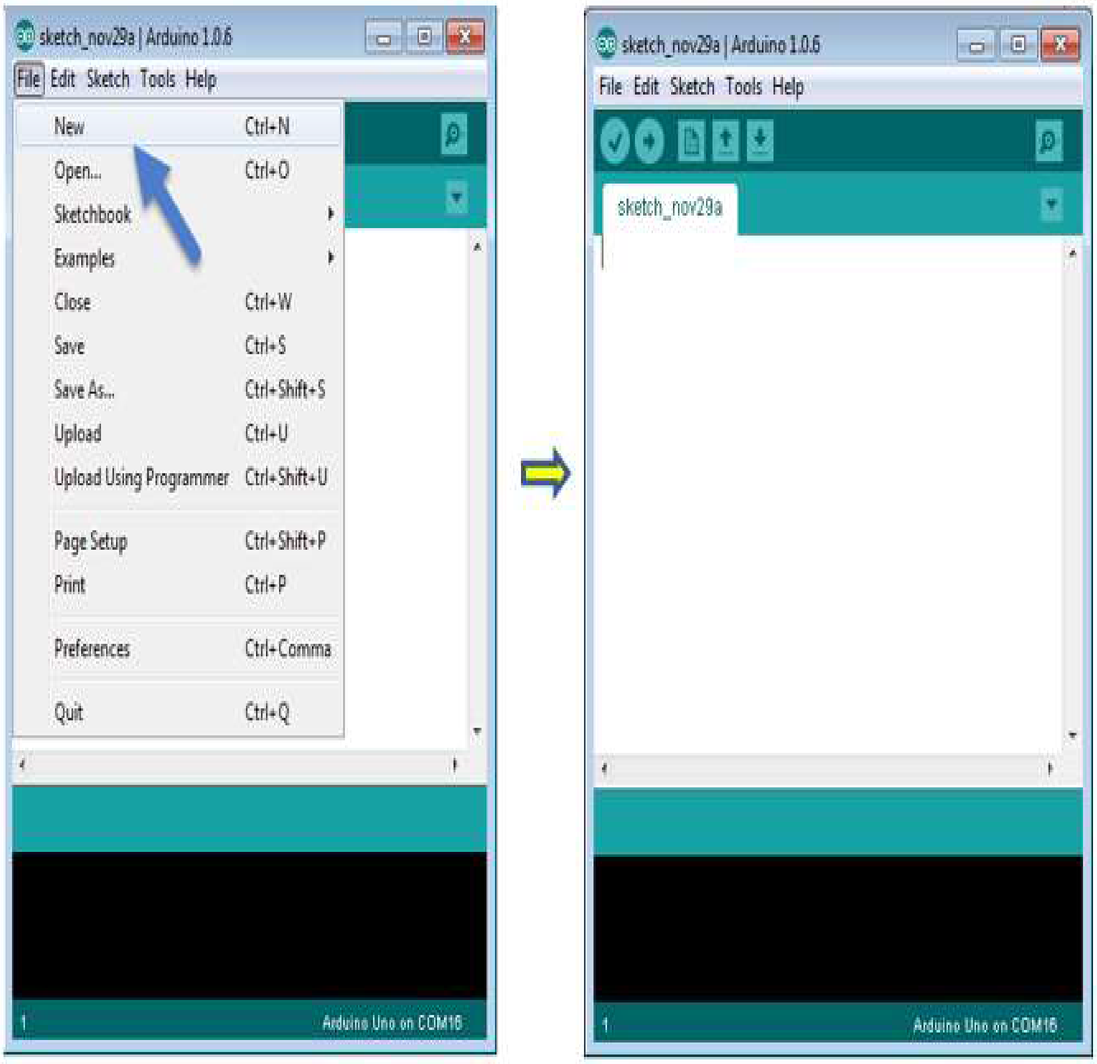


Figure.6.4 Creating a new project

To open an existing project example, select File -> Example -> Basics -> Blink.

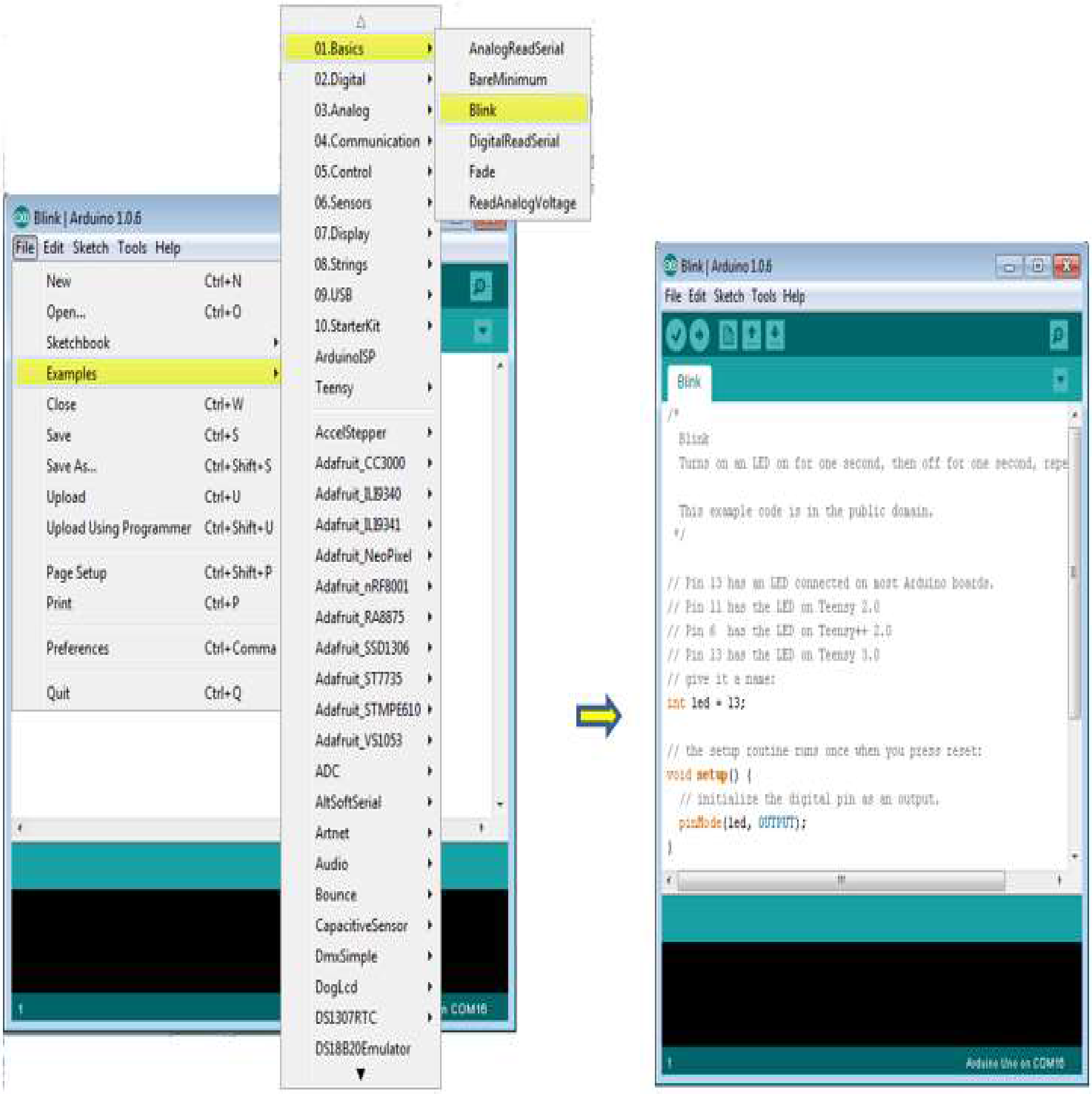


Figure .6.5 Opening of existing projects

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6: Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools -> Board and select your board

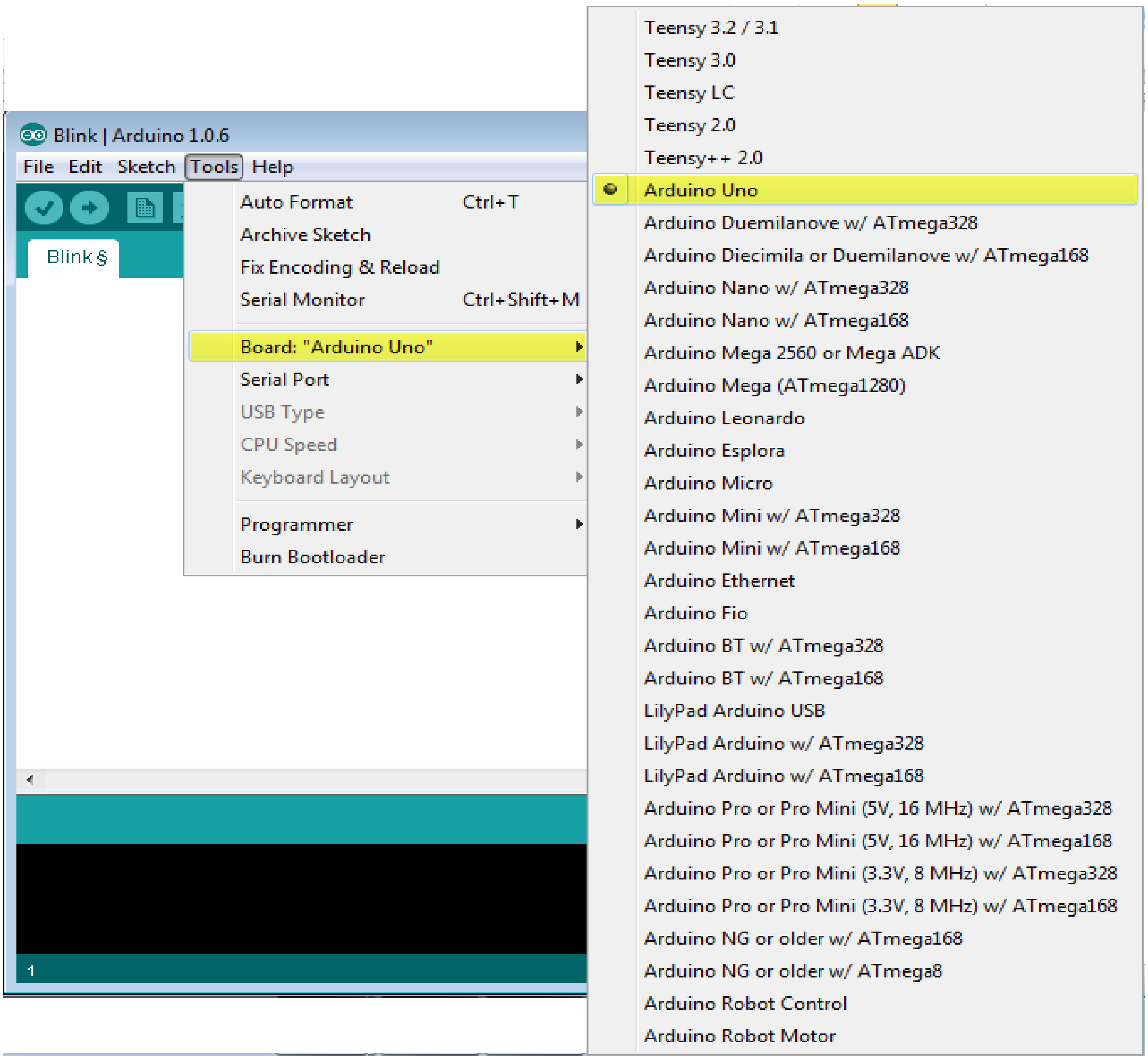


Figure. 6.6 Selecting Arduino Uno board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using

Step 7: Select your serial port.

Select the serial device of the Arduino board. Go to Tools ->Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

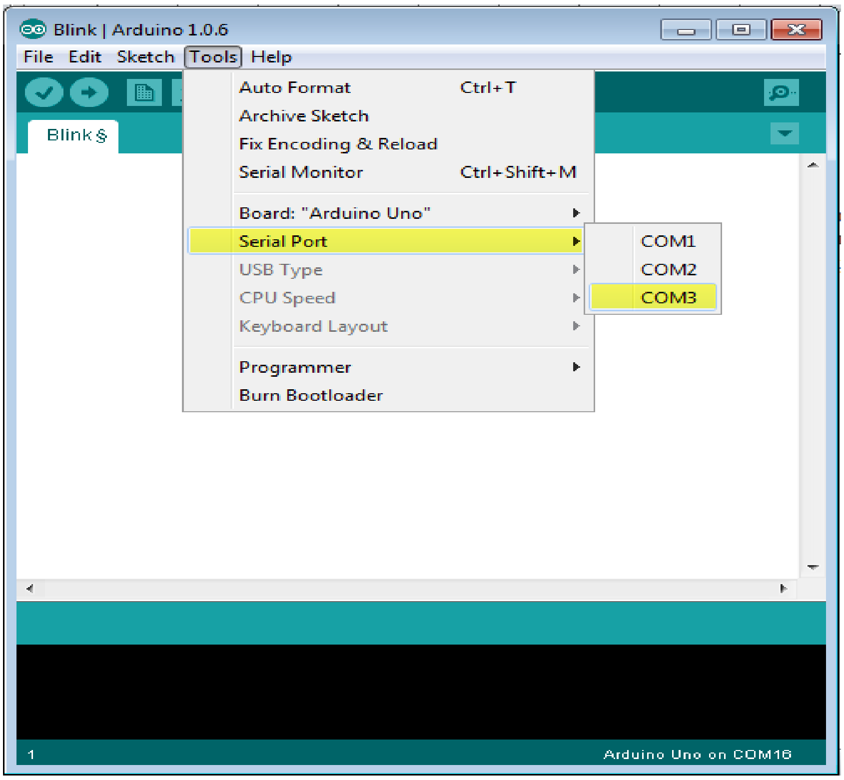


Figure. 6.7 Selecting serial port communication

Step 8: Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol app earing in the Arduino IDE toolbar.

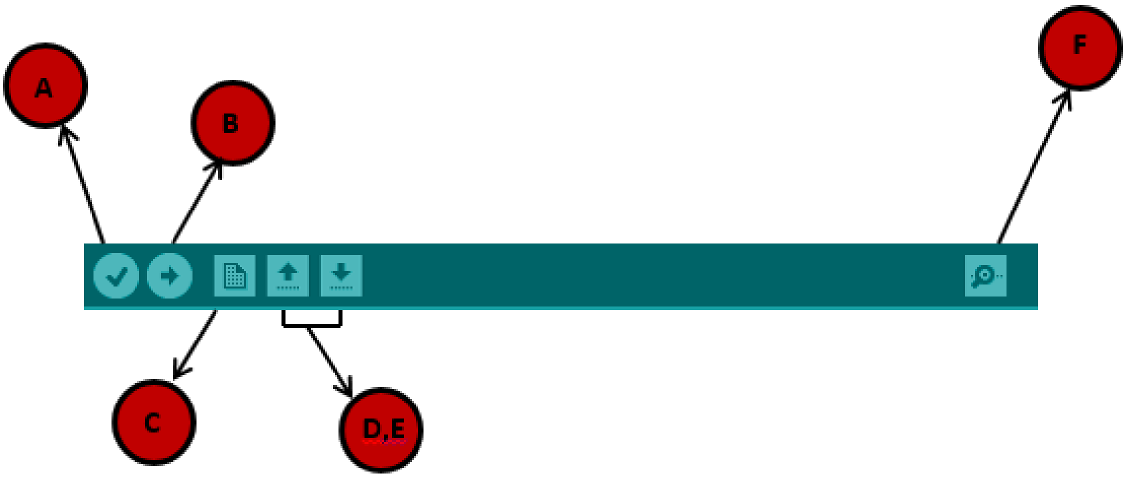


Figure. 6.8 Description of icons on new project window

A- Used to check if there is any compilation error.

B- Used to upload a program to the Arduino board.

C- Shortcut used to create a new sketch.

D- Used to directly open one of the example sketches.

E- Used to save your sketch.

F- Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Note: If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

6.1.2. ARDUINO PROGRAMMING STRUCTURE

In this chapter, we will study in depth, the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.

Sketch: The first new terminology is the Arduino program called “sketch”.

Structure

Arduino programs can be divided in three main parts: Structure, Values (variables and constants), and Functions. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.

Let us start with the Structure. Software structure consist of two main functions:

- Setup () function

- Loop () function

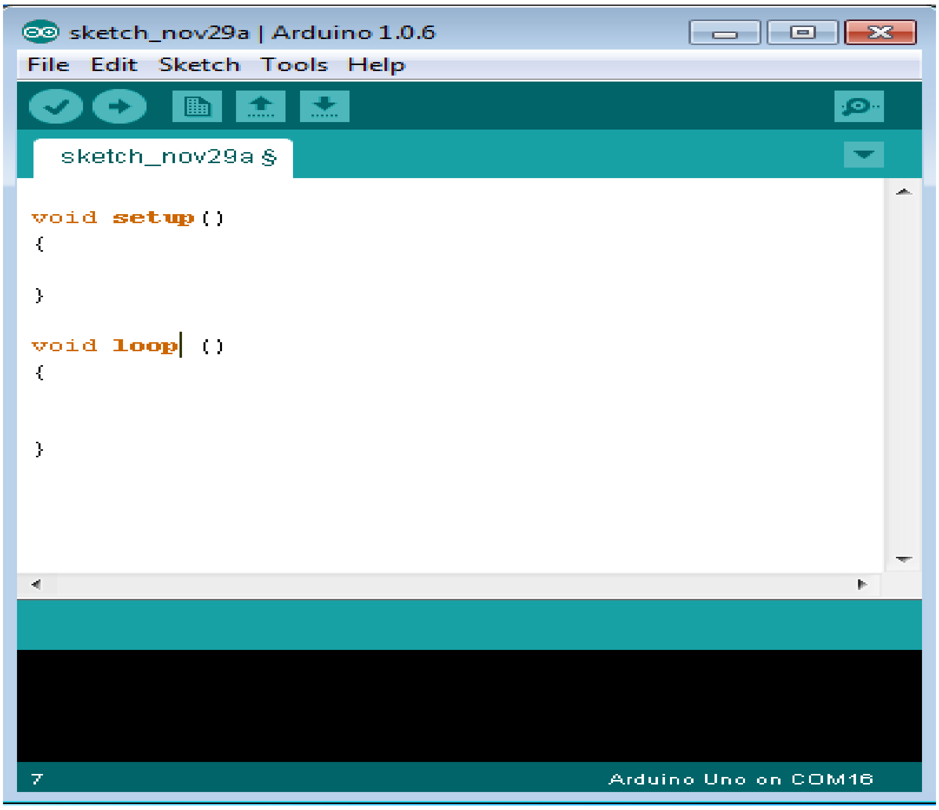


Figure 6.9 Starting window of new project

Void setup ()

{

}

PURPOSE:

The setup () function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

INPUT

OUTPUT

RETURN

```
Void Loop ()

{

}
```

PURPOSE:

After creating a setup () function, which initializes and sets the initial values, the loop () function does precisely what its name suggests, and loops secutively, allowing your program to change and respond. Use it to actively control the Arduino board.

CHAPTER-7

RESULT AND ANALYSIS

STEP-1

Proposed robot model on working condition

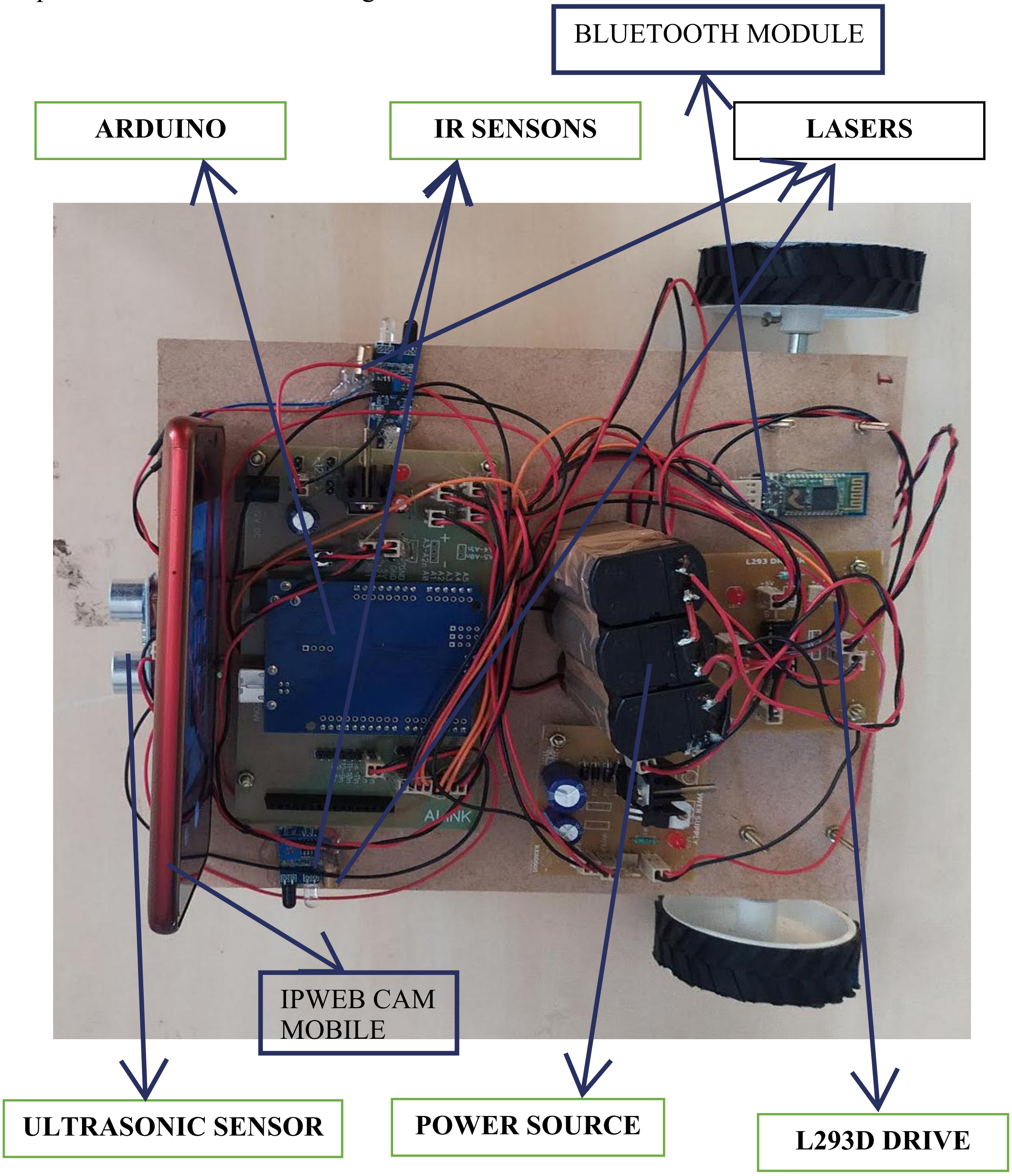


Figure 7.1 proposed robot model

STEP-2

Connecting robot to mobile with hc-05 Bluetooth module.

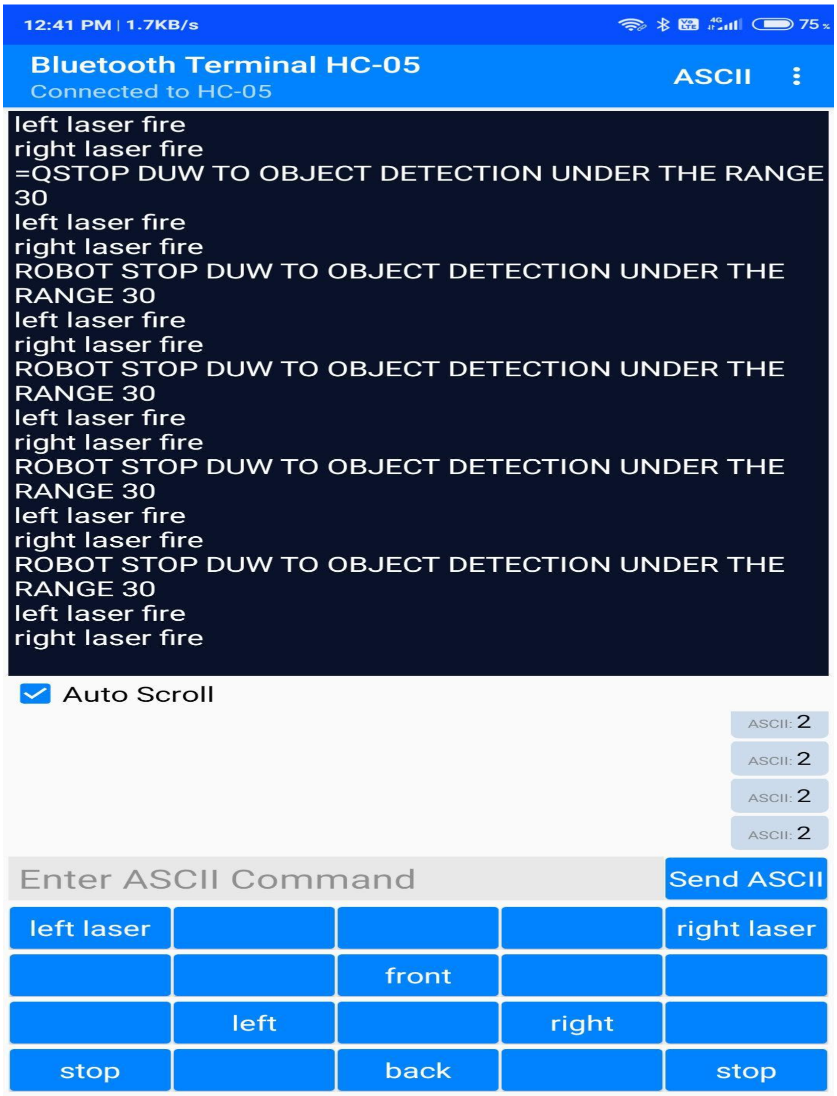


figure 7.2 hc-05 bluetooth control pannel and hc-05 device

STEP-3

Connecting web camera with robot to system.

Open IPWEBCAM application on mobile and click on start server. After starting server image display opens on that display IP address will found and copy that IP address and paste it in browser then IPDISPLAY OPEN

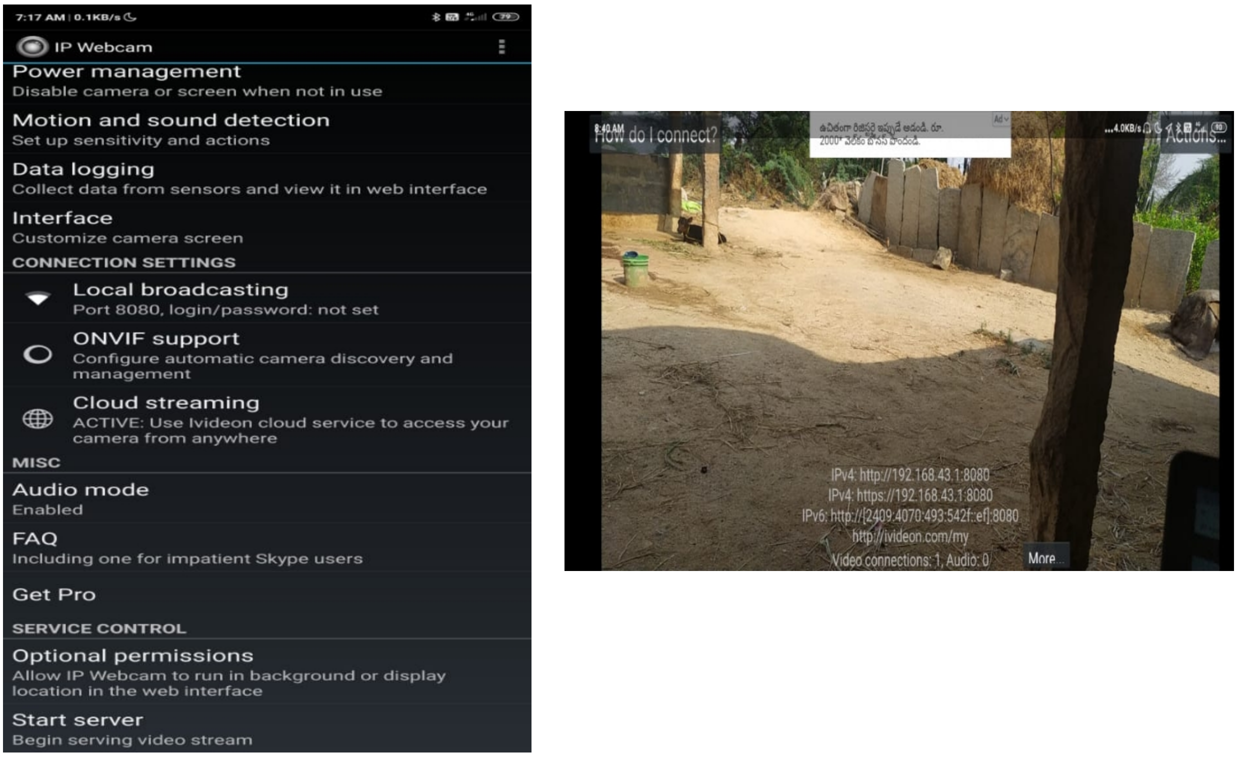


Figure 7.3 ipweb application and interface

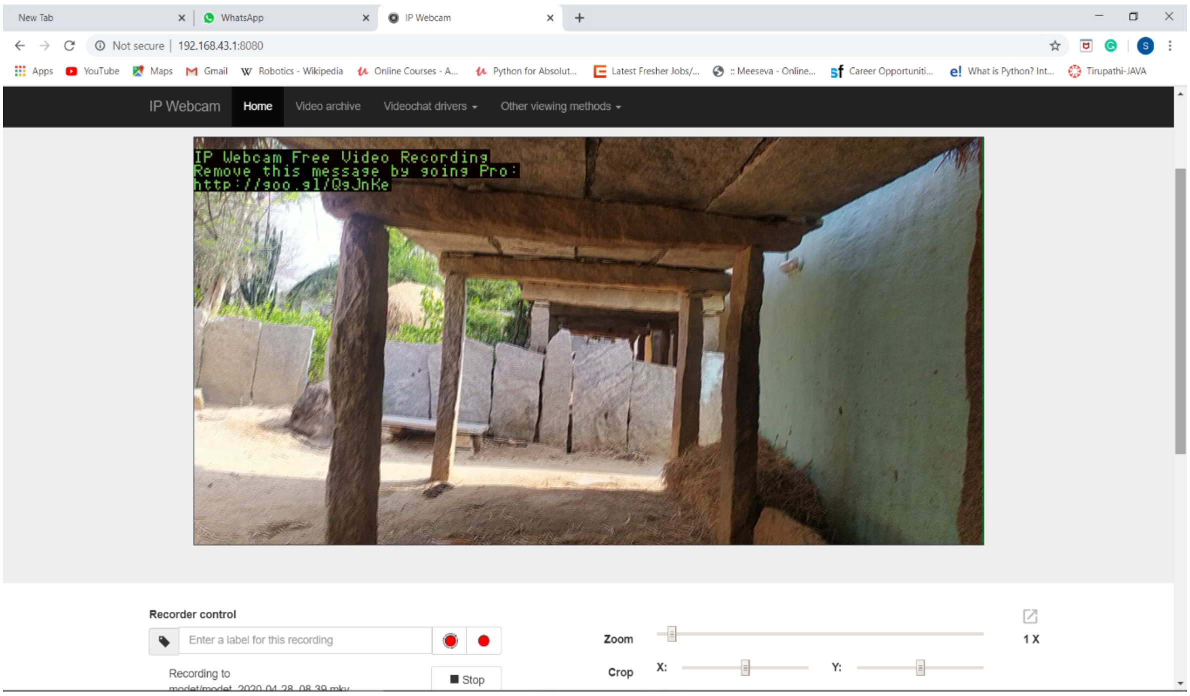


Figure 7.4 IPWEBCAM display on browser

STEP-4

OBSERVATION

Bluetooth controlled robot is controlled by using Android mobile phone. Here we use button in android phone to control the robot in forward, backward, left and right directions. So here android phone is used as transmitting device and Bluetooth module placed on the kit is used as receiver. Android phone will transmit command using its in-built Bluetooth, so that robot can move in the required direction like moving forward, reverse, turning left, turning right and stop.in this system we placed a laser to target the enemies, objects etc. When left IR sensors that present in the robotic system activates when an obstacle come besides to it and indicates with a left laser fire. When obstacle comes besides to the right IR sensor that present in the robotic system will activates and displays as right.

When an object that comes under below that the range that specified in the system that indicates that robot is stopped an obstacle will present will indicates.

Through ipweb cam we can see the objects that present Infront of the robot as live take decision what should be action that we should take. We can also take photos and video record for further investigation.

CHAPTER-8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

The future of IOT is virtually unlimited due to advances in technology and consumers' desire to integrate devices such as smart phones with household machines. Wi-Fi has made it possible to connect people and machines on land, in the air and at sea. It is critical that both companies and governments keep in ethics in mind as we approach the fourth Industrial Revolution. With so much data traveling from device to device, security in technology will be required to grow just as fast as connectivity in order to keep up with demands. Governments will undoubtable face tough decisions as to how far the private the sector is allowed to go in terms of robotics and information sharing. The possibilities are exciting, productivity will increase and amazing things will come by connecting the world.

By keeping the embedded devices in the environment for monitoring enables self-protection (i.e., smart environment) to the environment. To implement this need to deploy the sensor devices in the environment for collecting the data and analysis. By deploying sensor devices in the environment, we can bring the environment into real life i.e. it can interact with other objects through the network. Then the collected data and analysis results will be available to the end user through the Wi-Fi. The smart way to monitor environment and an efficient, low cost embedded system.

8.2 FUTURE SCOPE

Anything that touches consumer industry became a buzz word. So now Internet of Things is one among. What it is – the end goals are to bring all things we use in day to day life over network and can be accessed across the world over internet. That means every objects/gadget we use in a day to day life will have identify over network and its information can be consumes via Laptop, Tablet and mobile and including wearable like smart watches. Why we should do it – the first and foremost thing is automation. In a typical day, we all have 24 hours – 1/3 of time goes in bed, 1/3 of time goes in office/school and 1/3 third of time we have to spend for ourselves.

Smart Home:

you could automate stuff in home to make faster decision, communicate instantly, monitor the stuff which is most important – you can be more effective automating non-value added activities and spend more time where you want to focus (this is more of my personal productive advice when it comes to automating things). Home automation has a great potential for consumers market. Start-ups and Big companies joining together to take a leap forward

Smart Farm:

Country like India, we lose lot of money due to lack predictive systems in agriculture fields (excluding natural distastes). In today's environment we also lack the employees' engagement for agriculture fields. IOT has a great potential for smart farming – no matter where you located but you can still control things on field

Smart Transportation:

Already companies like Uber, OLA implemented a great connected platform, even IOT can be extended to our own transportation systems like cars, motorcycle and bikes etc. – every use case will have a different story. Think about you want to travel in a bus from your home town and your mobile phone alerts you the exactly time you should get out of your home.

Smart City:

I don't want to write about it this topic and there are enough articles in place to learn more about. Keeping city connected will help in many ways and make people more productive.

Smart Industry:

It applies to all industry that exists today; you can think of a new use-case every day to jump in Internet of Things world. It's quite common for every industry. I've also answered different questions related to IOT and you can refer them also follow IOT Geeks to learn more about Internet of Things. You will get some perspective for sure.

Adding of more sensors to monitor other environmental parameters such as soil PH sensor, CO2 and oxygen sensor while allowing the replacing of current sensors if a wider range of measurements is desired. And also, Integration of additional monitoring devices such as a WI-FI camera to monitor growth of agricultural product. And also, the data can be uploaded to web server continuously.

REFERENCES

1. Acevedo JJ, Arrue BC, Maza I, Ollero A (2014) A decentralized algorithm for area surveillance missions using a team of aerial robots with different sensing capabilities. In: IEEE International Conference on Robotics and Automation (ICRA), pp 4735–4740
2. Agmon N, Kraus S, Kaminka GA (2008) Multi-robot perimeter patrol in adversarial settings. In: IEEE International Conference on Robotics and Automation (ICRA), pp 2339–2345, DOI 10.1109/ROBOT.2008.4543563
3. Al Marzouqi M, Jarvis RA (2011) Robotic covert path planning: A survey. 2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM) pp 77–82, DOI 10.1109/RAMECH.2011.6070460
4. Alamdari S, Fata E, Smith SL (2013) Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations. International Journal of Robotics Research (IJRR) DOI 10.1177/0278364913504011
5. Alexander S, Bishop R, Ghrist R (2006) Pursuit and evasion in non-convex domains of arbitrary dimensions. In: Proceedings of Robotics: Science and Systems (RSS)
6. Dhiraj Arun Patil, Manish Y. Upadhye, F. S. Kazi, N. M. Singh, “Multi Robot Communication and Target Tracking System with Controller
7. Design and Implementation of SWARM Robot Using Arduino,” IEEE
8. International Conference on Industrial Instrumentation and Control
9. IEEE International Conference on Cyber Technology in Automation
10. Control and Intelligent Systems, May 27-31, 2012.