TABLE OF CONTENTS

Table of figure

Chapter 1: Abstract

People use their Smartphone for doing the regular tasks such as calling, texting, listening music and sending emails and other files. Now a days the Smartphone is not just used for talking purpose many other works are based on Smartphone such as web browser, games, and online videos. So with the help of this Smartphone user can direct the chair in four different directions. Android application is created and it is installed in the android Smartphone. The robot can move in two different modes.

First mode is touch mode as the touch screen is available in every Smartphone so it will be easy for user to use it. On the touch panel four different quadrants are given for LEFT, RIGHT, FORWARD and BACKWARD. The users just have to move his finger across the quadrant to select the direction and robot will move.

The second method is voice mode in this user just have to give the voice input to move the chair. Another advantage of this is sometimes the user is doing some work while driving in that case may the accident can happen to avoid such situation voice mode is given. Google API is already is built in mobile phone so that there is no need to add or install the other packages in the mobile phone. When the voice command is given it is converted into text-to- speech and the command goes to the microcontroller for execution.

The connection between android application and the robot is done using Bluetooth. The HC-05. Bluetooth module is used for this purpose. The user commands first given using application and then it is send to the microcontroller via Bluetooth. Bluetooth converts these commands in binary format send to the microcontroller. Then microcontroller execute the command and send the digital values to the device motor driver and at last device motor driver used to move the robot.

Chapter 2: Introduction

Robots are indispensable in many factoring industries. The reason is that the cost per hour to operate a robot is a fraction of the cost per hour to operate a human labor needed to perform the same function.

More than this, once programmed, robots repeatedly perform function with a high accuracy that surpasses that of the most experienced human operator. Human operators are, however, for more versatile. Humans can switch job tasks easily. Robots are built and programmed to be job specific. Robots are in the starting stage of their evolution. As robots evolve, they will become more versatile, emulating the human capacity and ability to switch job tasks easily. While the personal computer has made an indelible mark on society, the personal robot hasn't made an appearance. Obviously there's more to a personal robot than a personal computer robots require a combination of elements to be effective: sophistication of intelligence, movement, mobility, navigation and purpose.

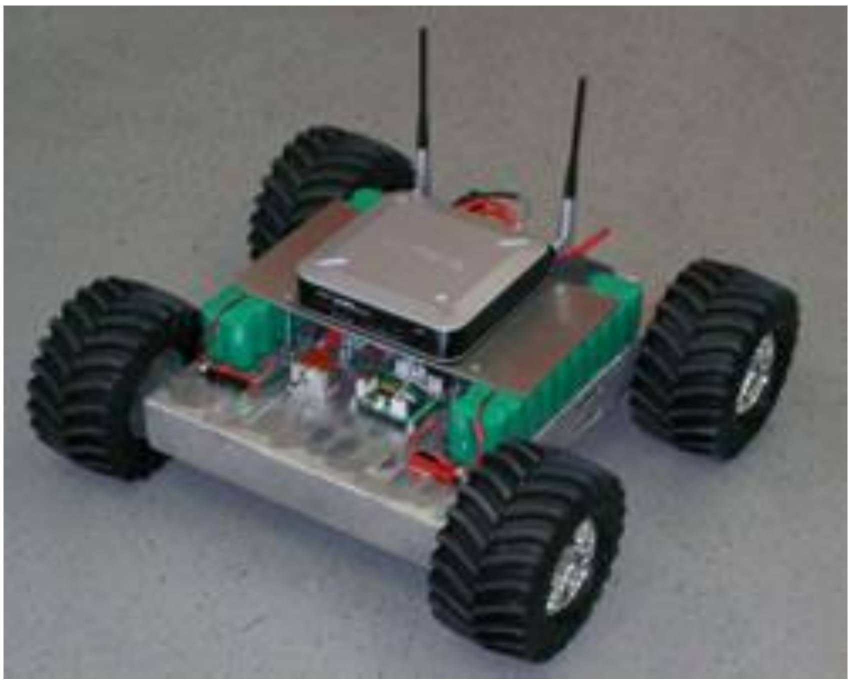

Fig 1: Wireless Robot

Chapter 3: Design

The Android application contains the functionality of handling the movement of robot and additional features like Emergency calling to the care taker and it can also send the emergency message to the intended person whose number is given when we starts the application. We have used incremental model to design application because if we want to make any changes in future then it will be easily accommodated. Incremental model should be used in such a project where requirements are well defines, but the realization will may be delayed.

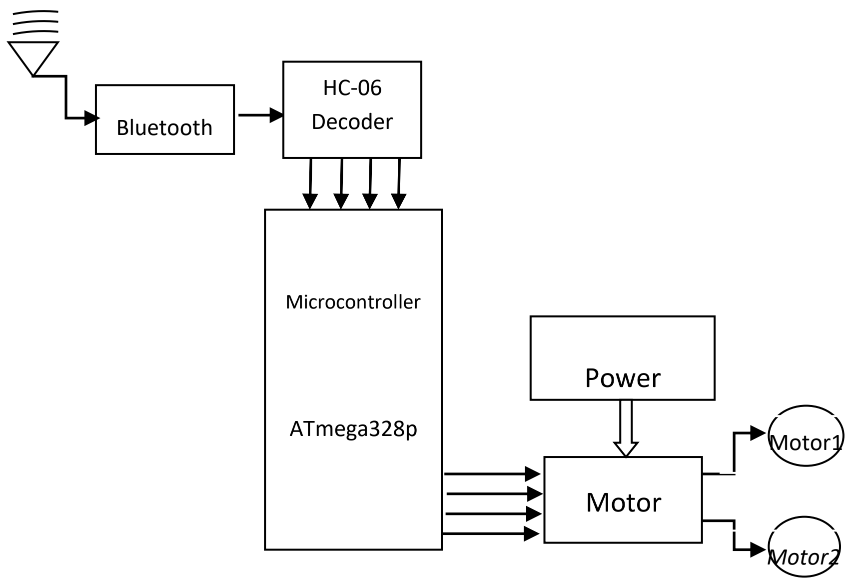

Fig 2: Basic block Diagram of the robot

In Incremental model the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycles take place here, making life cycle a “Multi-waterfall” cycle. Cycle are divided up into smaller, more easily managed module passed through requirements, design, implementation and testing phases A working version of software is produced during the first module. So we get working software early on during the Software life cycle [3]. Each subsequent release of module adds function to previous release. The process continues till the complete system is achieved. Incremental model is more flexible and less costly to change scope and requirements. In this model it is easy to test and debug during a smaller iteration.

PCB DESIGNING

The main purpose of printed circuit is in the routing of electric currents and signal through a thin copper layer that is bounded firmly to an insulating base material sometimes called the substrate. This base is manufactured with an integrally bounded layer of thin copper foil, which has to be partly etched or removed to arrive at a pre-designed pattern to suit the circuit connections, or other applications as required. The term printed circuit board is derived from the original method where a printed pattern is used as the mask over wanted areas of copper. The PCB provides an ideal baseboard upon which to assemble and hold firmly most of the small components.

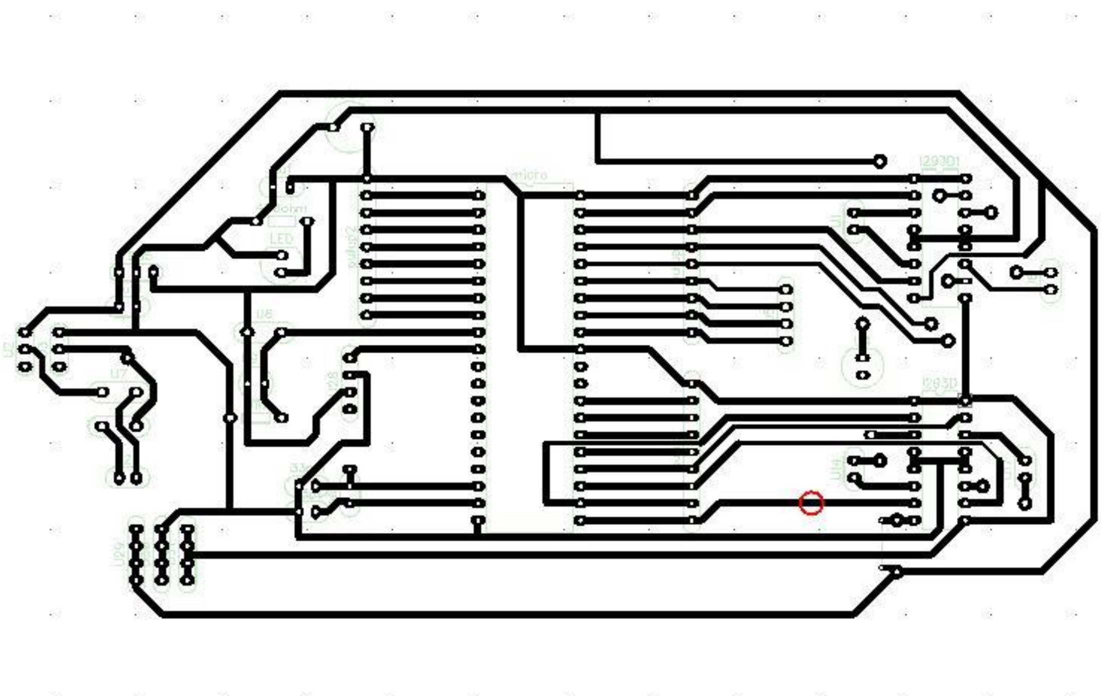

Fig 3: PCB LAYOUT 1 OF ROBOT

From the constructor's point of view, the main attraction of using PCB is its role as the mechanical support for small components. There is less need for complicated and time consuming metal work of chassis contraception except perhaps in providing the final enclosure. Most straight forward circuit designs can be easily converted in to printed wiring layer the thought required to carry out the inversion cab footed high light an possible error that would otherwise be missed in conventional point to point wiring .The finished project is usually neater and truly a work of art. Actual size PCB layout for the circuit shown is drawn on the copper board. The board is then immersed in FeCl3 solution for 12 hours. In this process only the exposed copper portion is etched out by the solution.Now the petrol washes out the paint and the copper layout on PCB is rubbed with a smooth sand paper slowly and lightly such that only the oxide layers over the Cu are removed. Now the holes are drilled at the respective places according to component layout as shown in figure.

LAYOUT DESIGN

When designing the layout one should observe the minimum size (component body length and weight). Before starting to design the layout we need all the required components in hand so that an accurate assessment of space can be made. Other space considerations might also be included from case to case of mounted components over the printed circuit board or to access path of present components. It might be necessary to turn some components around to a different angular position so that terminals are closer to the connections of the components. The scale can be checked by positioning the components on the squared paper. If any connection crosses, then one can reroute to avoid such condition.

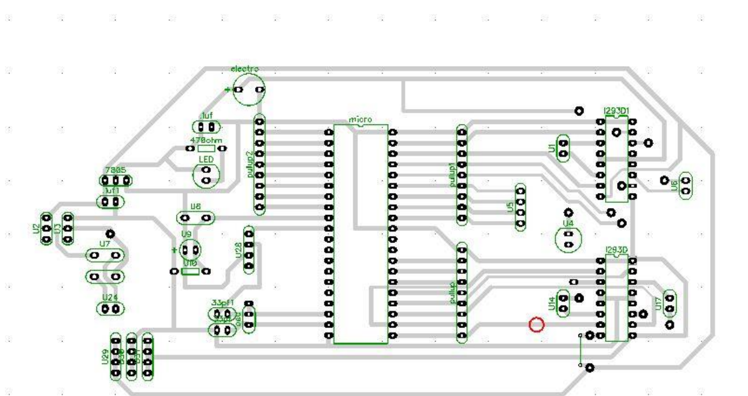

Fig 4: PCB Layout 2 of Robot:

All common or earth lines should ideally be connected to a common line routed around the perimeter of the layout. This will act as the ground plane. If possible try to route the outer supply line to the ground plane. If possible try to route the other supply lines around the opposite edge of the layout through the centre. The first set is tearing the circuit to eliminate the crossover without altering the circuit detail in any way.Plan the layout looking at the topside to this board. First this should be translated inversely; later for the etching pattern large areas are recommended to maintain good copper adhesion. It is important to bear in mind always that copper track width must be according to the recommended minimum dimensions and allowance must be made for increased width where termination holes are needed. From this aspect, it can become little tricky to negotiate the route to connect small transistors.

There are basically two ways of copper interconnection patterns under side the board. The first is the removal of only the amount of copper necessary to isolate the junctions of the components to one another. The second is to make the interconnection pattern looking more like conventional point wiring by routing uniform width of copper from component to component.

Chapter 4: Circuit Diagram of PCB design

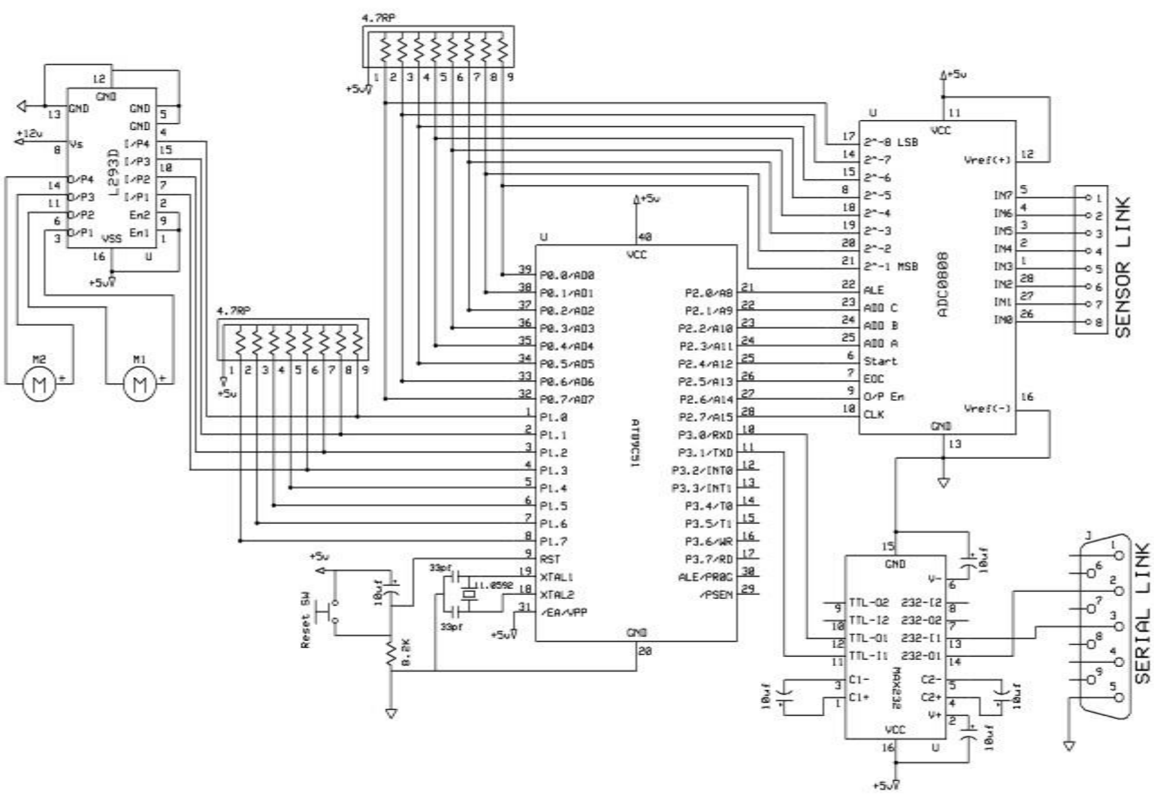

Fig 5: Circuit Diagram of PCB design

Chapter 5: Robotic Arm

The new design of the arm of the robot should enable the robot to make the same movements as a human arm. Also it should be able to make those movements while carrying a weight of approximately 0.5 kg. In order to simulate this, a trajectory is chosen. In this trajectory the arm of the robot will move to an object on a table which is directly in front of the robot and approximately at abdominal height. This trajectory is shown in Figure 5.

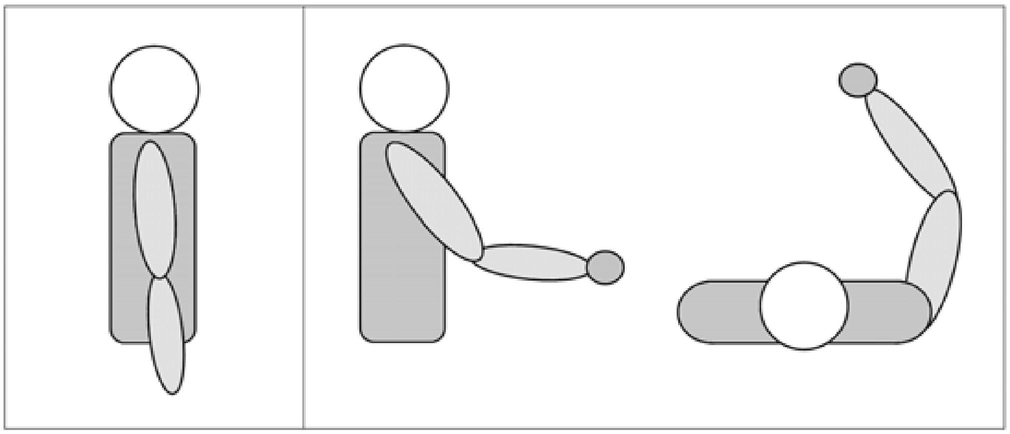

Fig 5: Human arm movement

In order to let the arm move in a human like way the arm should have more degrees of freedom (DOF's)
than the robot has at this moment. Currently the robot has one DOF in the shoulder which allows the robot to swing the whole arm sideways to the body.

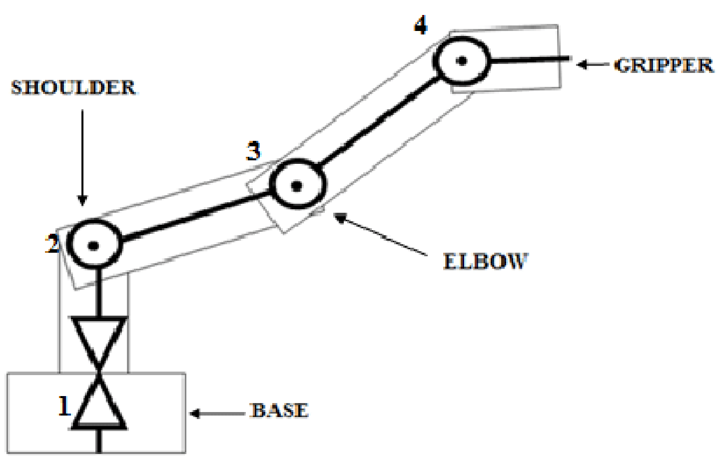

Fig 7: Robotic arm movement

Chapter 6: Methodology

In this project Android application is connected to the car via Bluetooth. For connection the HC-05 module is used inside Microcontroller.

The figure 2 shows the system architecture which defines the actual working of system. We have provided the IR sensors which helps to avoid accidents happens due to any obstacle. User can use two features provided in application either Voice or Touch mode and these commands will be forwarded to the Microcontroller mounted on car via Bluetooth. We have used battery as power supply which helps to accommodate large distance.

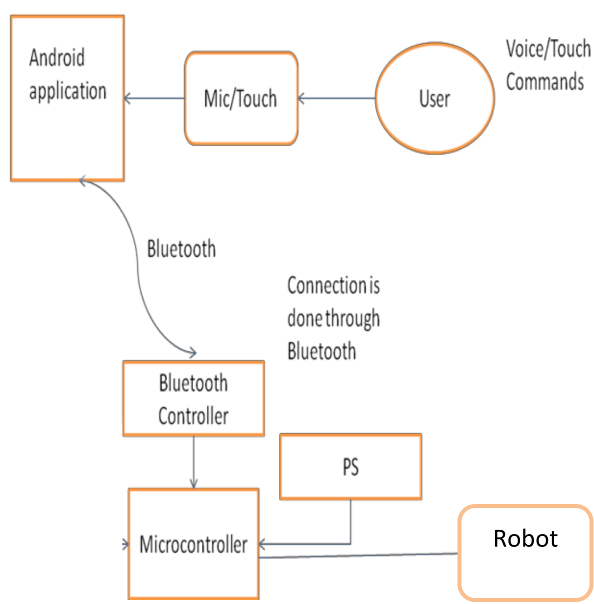

Fig 8: System Architecture

For implementation of hardware part we have used four ICs these are: 89C51 the main microcontroller, L293D the DC motor driver, MAX232 for serial communication and ADC0808 for analog to digital convertor. 89C51 is 40 pin IC which has 4K byte of memory and it has less cost.

The work is mainly divided in three modules in which first are: creation of application and User interaction with application. This module creates Graphical Interface for any non-technical user can also interact with application. It will also provides options like select different modes of controlling wheel chair like (Voice and Touch commands) for controlling. In this microcontroller maintains unique information of commands and match that commands with user input and perform action like moving Left, Right, Forward, Reverse, Stop. The second module is Connection of application to Microcontroller. In this Module Android application is started first and then it is connected to the Microcontroller via the Bluetooth. After starting the Bluetooth we will get the list of devices near available and need to pair with the Microcontroller and then after connecting it becomes ready to perform action on car. And third module is Connection of Microcontroller to the Wheel chair. This module describes the working of microcontroller and actual connection with Device Driver and car. After forming connection with application, microcontroller initializes the connection with device driver (i.e. Left and right side Motor) and after taking the input from user it will send that input in the form of bits to the Device driver[3,4].

Sensors are attached to the microcontroller they work as follows: Sensors are the integral part of the system. This helps to detect whether there is any obstacle present or not. Sensor detects obstacle by sending continuous signal from transmitter and if there is obstacle then that signal will be reflected back to the Receiver. If receiver receive signal then there is obstacle and it will notify about this to the user and this helps to avoid accidents.

Chapter 7: Component Description & Working

In this project we are using many components. Here we have mentioned some major components of our project. The description of those components is given below-

ATmega-328 Microcontroller

ATmega-328 is basically an Advanced Virtual RISC (AVR) micro-controller. It supports the data up to eight (8) bits. ATmega-328 has 32KB internal built-in memory. ATmega 328 has 1KB Electrically Erasable Programmable Read Only Memory (EEPROM). This property shows if the electric supply supplied to the micro-controller is removed, even then it can store the data and can provide results after providing it with the electric supply. It operates ranging from 3.3V to from its biasing terminals. Its excellent features include the cost efficiency, low power dissipation, programming lock for security purposes, real timer counter with separate oscillator. It's normally used in Embedded Systems applications. ATmega-328 is shown in the figure given below.

Fig 9: Atmega328 IC

1. ATmega328 Pins

- ATmega-328 is an AVR Microcontroller having twenty eight (28) pins in total.
- All of the pins in chronological order, are listed in the table shown in the figure given below.

ATmega328 Pins			
Pin Number	**Pin Name**	**Pin Number**	**Pin Name**
1	PC6	15	PB1
2	PD0	16	PB2
3	PD1	17	PB3
4	PD2	18	PB4
5	PD3	19	PB5
6	PD4	20	AVCC
7	V_{CC}	21	A_{REF}
8	GND	22	GND
9	PB6	23	PC0
10	PB7	24	PC1
11	PD5	25	PC2
12	PD6	26	PC3
13	PD7	27	PC4
14	PB0	28	PC5

Table 1: Pins in chronological order

2. ATmega328 Pinout

- Through pinout diagram we can understand the configurations of the pins of any electronic device, so you are working on any Engineering Project then you must first read the components' pinout.
- ATmega 328 pinout diagram is shown in the figure given below.

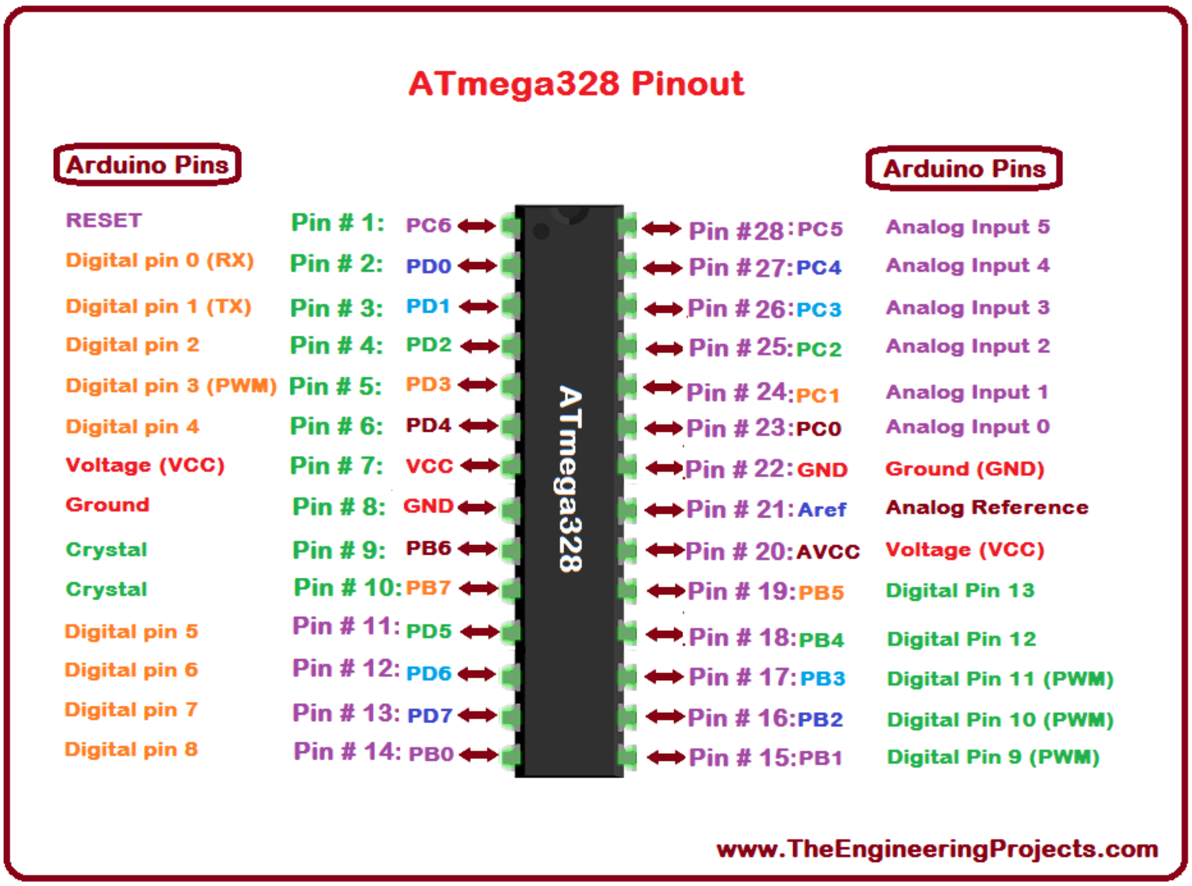

Fig 10: Pinout diagram

3. ATmega328 Pins Description

- Functions associated with the pins must be known in order to use the device appropriately.
- ATmega-328 pins are divided into different ports which are given in detail below.

VCC is a digital voltage supply.
AVCC is a supply voltage pin for analog to digital converter.
GND denotes Ground and it has a 0V.
Port A consists of the pins from **PA0** to **PA7.** These pins serve as analog input to analog to digital converters. If analog to digital converter is not used, **port A** acts as an eight (8) bit bidirectional input/output port.
Port B consists of the pins from **PB0** to **PB7.** This port is an 8 bit bidirectional port having an internal pull-up resistor.
Port C consists of the pins from **PC0** to **PC7.** The output buffers of **port C** has symmetrical drive characteristics with source capability as well high sink.
Port D consists of the pins from **PD0** to **PD7.** It is also an 8 bit input/output port having an internal pull-up resistor.

- All of the AVR ports are shown in the figure given below.

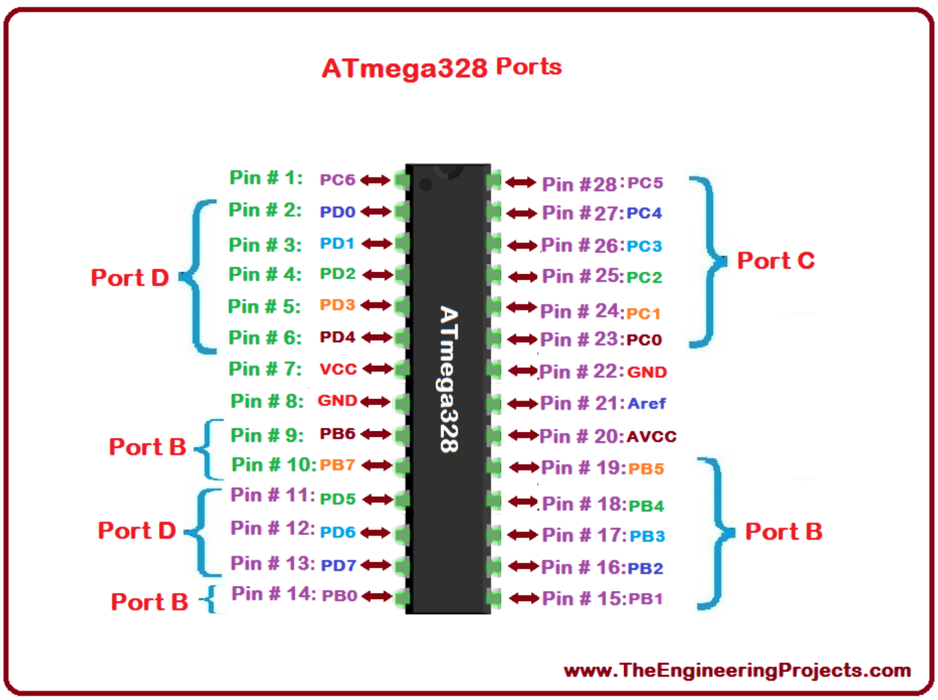

Fig 11: Pin description

4. ATmega328 Architecture

- Architecture of a device presents each information about the particular device.
- ATmega-328 architecture is shown in the figure given below.

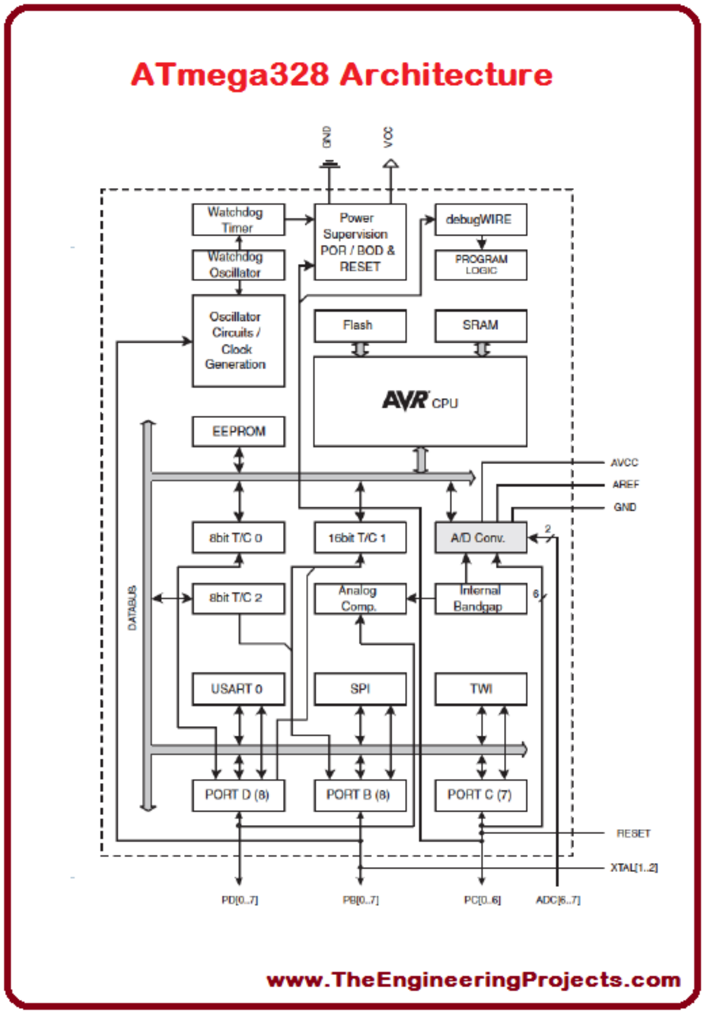

Fig 12: Architecture of ATmega 328

5. ATmega328 Memory

- ATmega 328 has three types of memories e.g. EEPROM, SRAM etc.
- The capacity of each memory is explained in detail below.

Flash Memory has 32KB capacity. It has an address of 15 bits. It is a Programmable Read Only Memory (ROM). It is non volatile memory.
SRAM stands for Static Random Access Memory. It is a volatile memory i.e. data will be removed after removing the power supply.
EEPROM stands for Electrically Erasable Programmable Read Only Memory. It has a long term data.

- AVR memory spaces are shown in the figure given below.

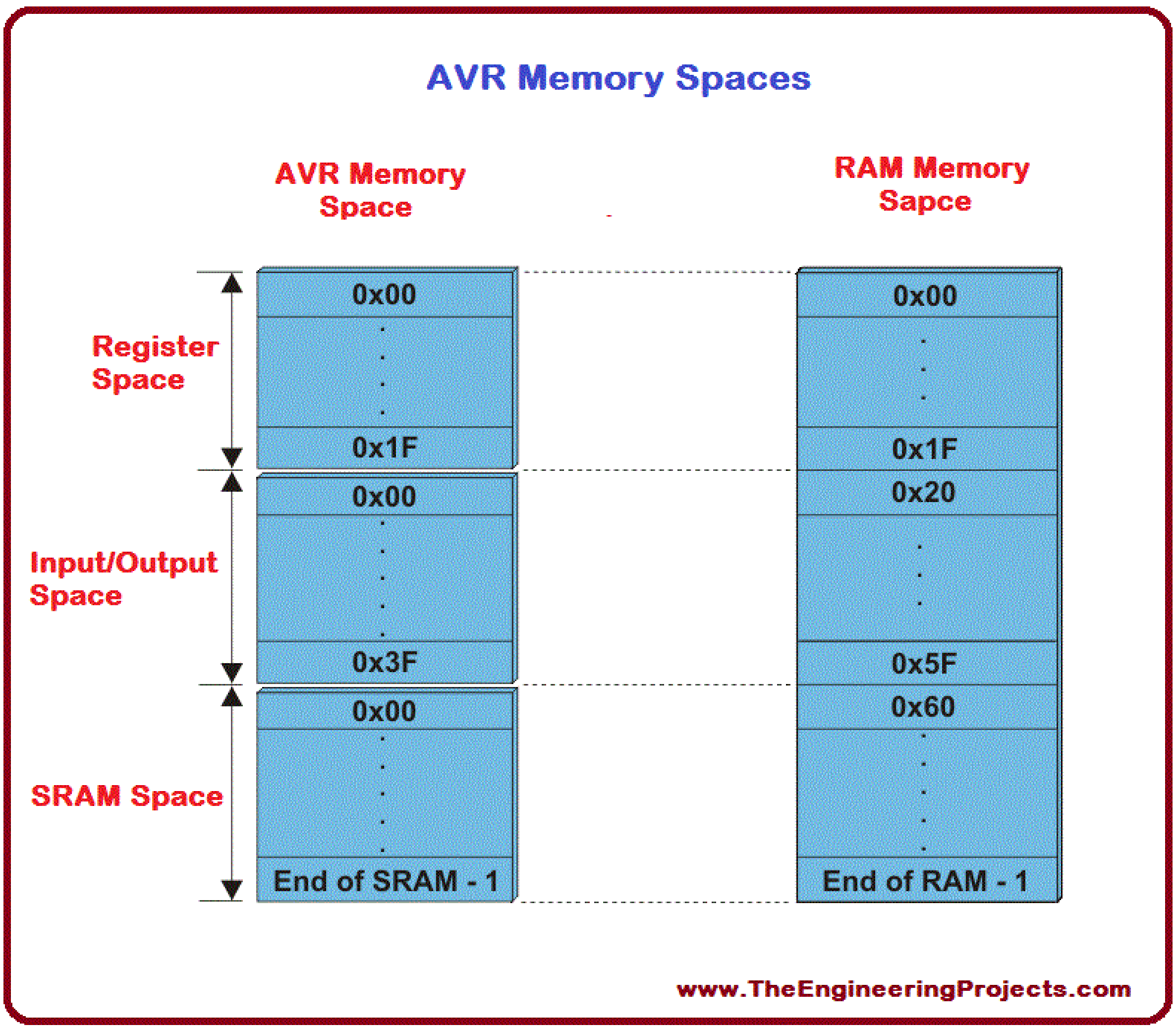

Fig 13: AVR space

6. ATmega328 Registers

- ATmega-328 has thirty two (32) General Purpose (GP) registers.
- These all of the registers are the part of Static Random Access Memory (SRAM).
- All the registers are given in the figure shown below.

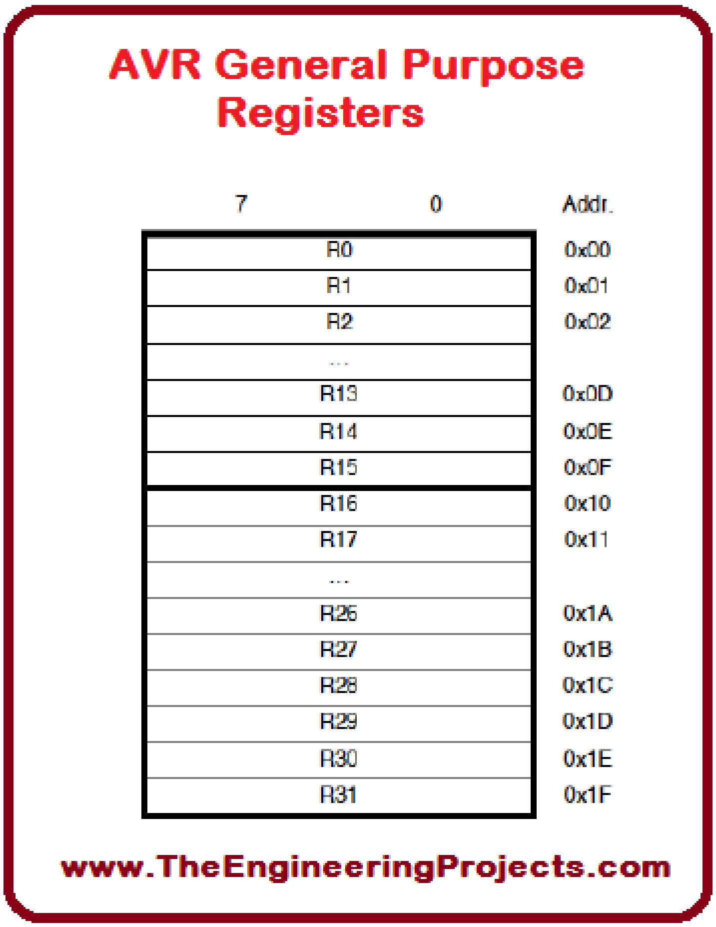

Fig 14: ATmega 328 registers

7. ATmega328 Packages

- The different versions of the same device are denoted by the different packages of that device.
- Each package has different dimensions, in order to differentiate easily.
- ATmega 328 packages are given in the table shown in the figure given below.

ATmega328 Package	
Part. No	**Packages**
ATmega328-AU	32A
ATmega328-AUR	32A
ATmega328-MMH	28M1
ATmega328-MU	321M1-A
ATmega328-PU	28P3

Table 2: ATmega 328 packages

8. ATmega328 Block Diagram

- Block diagram shows the internal circuitry and the flow of the program of any device.
- ATmega 328 block diagram is shown in the figure given below.

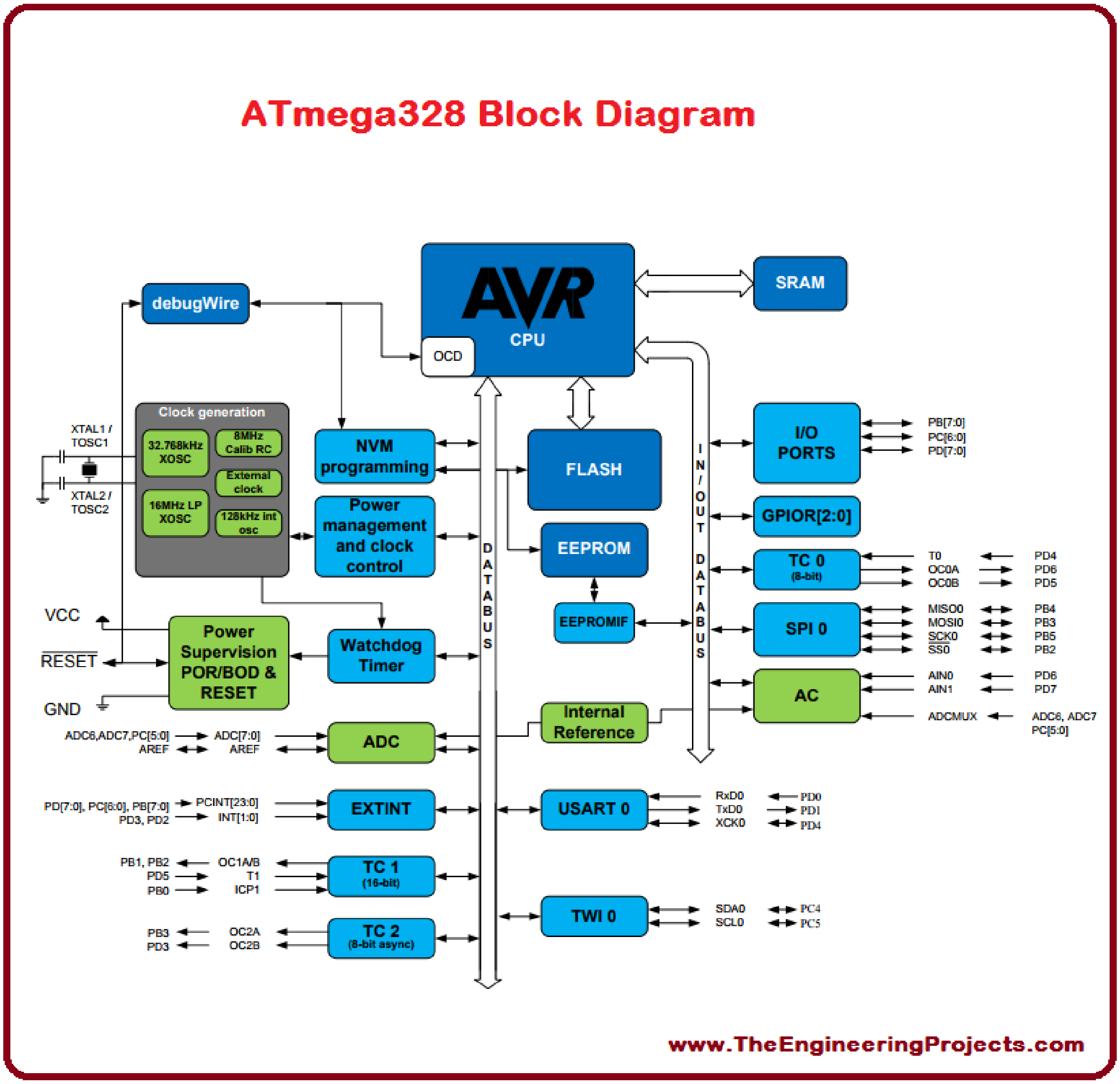

Fig 15: ATmega internal circuitry

9. ATmega328 Features

- To perform any task we can select a device on the basis of its features. i.e whether its features match to obtain the desired results or not.
- Some of the main features of an AVR Microcontroller ATmega328 are shown in the table given in the figure below.

ATmega328 Features	
Sr. No	**Features**
1	Non programmable data and program memory
2	High performance
3	Low power consumption
4	Fully static operation
5	On chip analog comparator
6	Advance RISC architecture
7	32KB flash memory
8	2KB SRAM

Table 3: Features of an AVR Microcontroller

10. ATmega328 and Arduino

- ATmega-328 is the most micro-controller that is used while designing.
- ATmega 328 is the most important part of Arduino.
- The program is uploaded on the AVR micro-controller attached on Arduino.
- AVR attached on Arduino is shown in the figure given below.

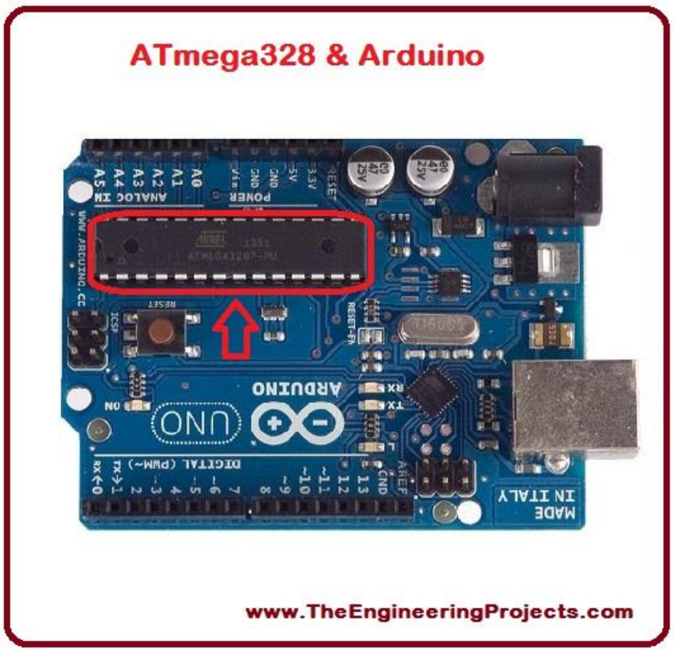

Fig 16: AVR attached on Arduino

11. ATmega328 and Arduino Pins

- ATmega328 pins are connected to the corresponding pins of Arduino.
- Their connectivity with each other is shown in the pinout diagram shown in the figure given below.

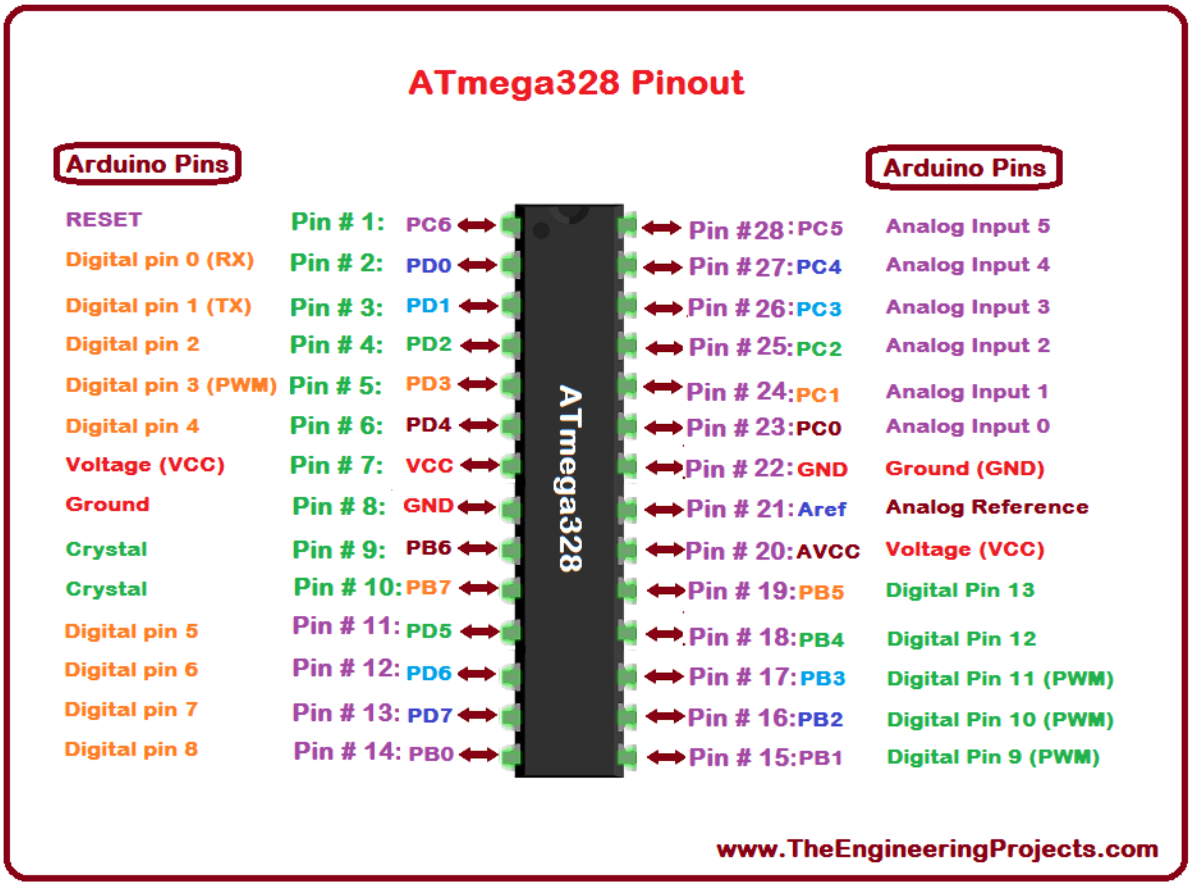

Fig 17: Connectivity of ATmega328

- The encircled section analog pins consist of the Arduino pins which are connected to the corresponding AVR micro-controller ATmega-328 pins.
- I have written both of the pins in front of each other, it will help to understand easily.
- If you want to work on this Arduino board then you must try these Arduino Projects for Beginners, they will help to get your hands on Arduino.

12. Applications

- A complete package including ATmega 328 and Arduino can be used in several different real life applications.
- It can be used in Embedded Systems Projects.
- It can also be used in robotics.
- Quad-copter and even small aero-plane can also be designed through it.
- Power monitoring and management systems can also be prepared using this device.

13. HC05-TTL Bluetooth Transceiver

Description: The Bluetooth Transceiver HC-05 Breakout is the latest Bluetooth wireless serial cable. This version of the popular Bluetooth uses the HC-05/HC-06 module. These modems work as a serial (RX/TX) pipe. Any serial stream from 9600 to 115200bps can be passed seamlessly from your computer to your target. The remote unit can be powered from 3.3V up to 6V for easy battery attachment. All signal pins on the remote unit are 3V-6V tolerant. No level shifting is required. Do not attach this device directly to a serial port. You will need an RS232 to TTL converter circuit or Arduino ZigBee USB Adapter if you need to attach this to a computer. You can either solder a 6-pin header or individual wires. Unit comes without a connector. And now, we provide HC-05/06, HC-05 could be setting to Master or Slave by user. HC-06 just be Master or Slave that could be customized.

14. Key Features:

1. CSR Bluetooth Chip Solution

2. Bluetooth Spec v2.0+EDR Compliant

3. Enhanced Data Rate (EDR) compliant with V2.0.E.2 of specification for both 2 Mbps and 3 Mbps modulation modes

4. Full Speed Bluetooth Operation with Full Piconet Support and Scatternet Support

5. Incredible small size with 3.3V input, and RoHS Compliant

6. UART interface and with baud rate setup function

7. Support for 8Mbit External Flash Onboard

8. Support for 802.11Co-Existence

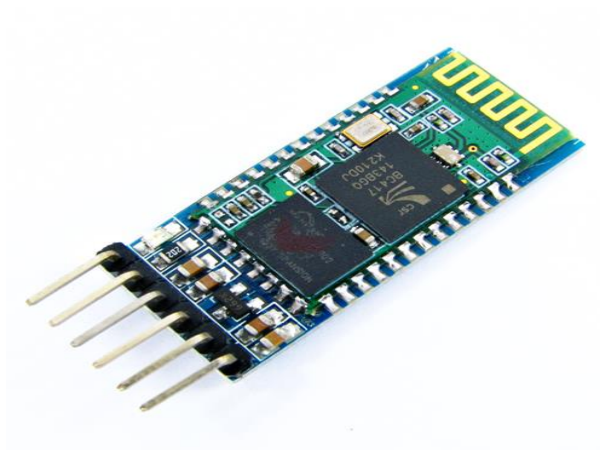

Fig 18: HC-05 Bluetooth Module

15. D.C. Motor

An electric motor is a machine which converts electrical energy into mechanical energy.

Principle:

It is based on the principle that when a current-carrying conductor is placed in a magnetic field, it experiences a mechanical force whose direction is given by Fleming's Left-hand rule and whose magnitude is given by

Force, **F = B I L** newton

Where B is the magnetic field in weber/m^2.

I is the current in amperes and

L is the length of the coil in meter.

The force, current and the magnetic field are all in different directions.

If an Electric current flows through two copper wires that are between the poles of a magnet, an upward force will move one wire up and a downward force will move the other wire down.

The loop can be made to spin by fixing a half circle of copper which is known as commutator, to each end of the loop. Current is passed into and out of the loop by brushes that press onto the strips. The brushes do not go round so the wire do not get twisted. This arrangement also makes sure that the current always passes down on the right and back on the left so that the rotation continues. This is how a simple Electric motor is made.

Fig 19: DC Motor

Here we have mentioned some basic components of our project-

16. **Power Supply/Power Adopter:**

Power supply is the source of electrical power. Normally we use +5V DC power for regular working of almost any electronic circuit. User can directly built +5V DC power supply using 4 diodes, filter capacitors and regulator IC - 7805 (Integrated Circuit) or can directly purchase a +5V DC power adopter from the local market

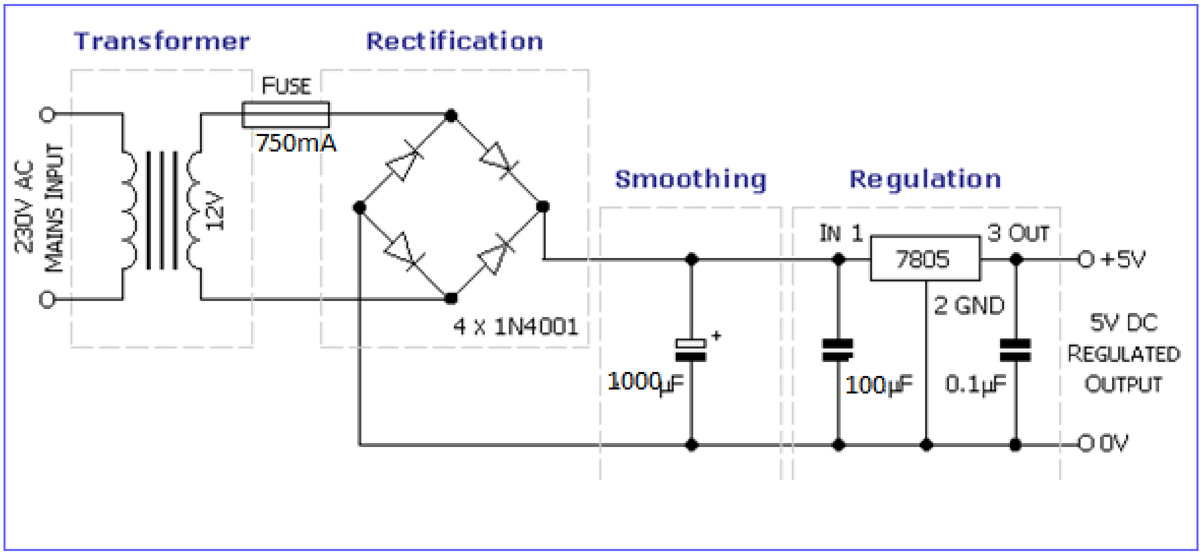

Fig 20: Power Supply

17. **7805 voltage regulator**:

It is an integrated circuit. It is a member of 78xx series of fixed linear voltage regulator ICs. The voltage source in a circuit may have fluctuations and would not give the fixed voltage output.

The primary and the most important part of an electronic circuit either analog or digital is a power supply. A block that fulfills the necessary power i.e. the voltage and current requirement of the circuit.

A Voltage Regulator is a semiconductor device that converts an input DC voltage to a fixed output DC voltage. 78XX series of IC regulators is representative of three terminal devices that are available several fixed positive output voltages making them useful in a wide range of applications. It has three terminals labelled as input, output, and ground. The last two digits designate the output voltage. For example, IC 7805 is a +5v voltage re

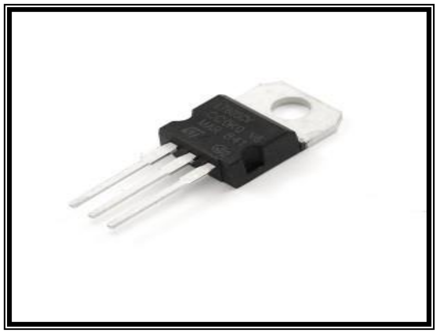

Fig 21: IC 7805

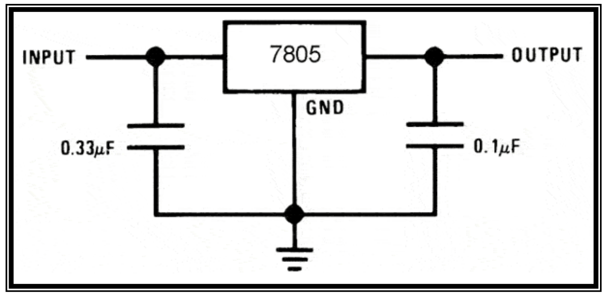

Fig 22: Block diagram of voltage regulator

Pin No	**Function**	**Name**
1	Input voltage (5V-18V)	Input
2	Ground (0V)	Ground
3	Regulated output; 5V (4.8V-5.2V)	Output

Table 5: Pin descriptions of 7805

18. Resistor:

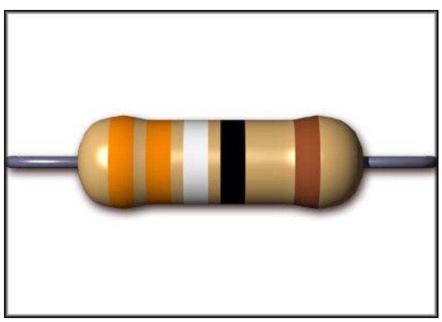

Fig 23: Resistor

A resistor is a two-terminal passive electronic component which implements electrical resistance as a circuit element. When a voltage V is applied across the terminals of a resistor, a current I will flow through the resistor in direct proportion to that voltage. This constant of proportionality is called conductance, G. The reciprocal of the conductance is known as the resistance R, since, with a given voltage V, a larger value of R further "resists" the flow of current I as given by

$$I = \frac{V}{R}$$

Units:

The Ohm (symbol: Ω) is the SI unit of electrical resistance, named after Georg Simon Ohm. An ohm is equivalent to a volt per ampere. Since resistors are specified and manufactured over a very large range of values, the derived units of milliohm (1 mΩ = 10^{-3} Ω), kilo ohm (1kΩ = 10^{3} Ω), and mega ohm (1 MΩ = 10^{6} Ω) are also in common usage.

19. Capacitor

Fig 24: Electrolytic Capacitor and Capacitor Symbol

A capacitor (formerly known as condenser) is a device for storing electric charge. The forms of practical capacitors vary widely, but all contain at least two conductors separated by a non-conductor. Capacitors used as parts of electrical systems, for example, consist of metal foils separated by a layer of insulating film.

20. Bridge Rectifier:

To drive the current of 10A from the transformer we need to have rectifier that can bear such a heavy current rating. For that purpose we made a bridge rectifier by four diodes. The diode used was 6A4.

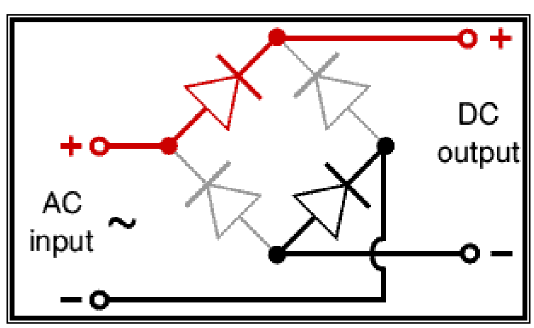

Fig 25: Block diagram of bridge rectifier

Also we used here GD W10M bridge rectifier which is used in microcontroller circuit. We had to use here to different bridge rectifier from two different power supplies because power requirement of motor is large enough to microcontroller circuit.

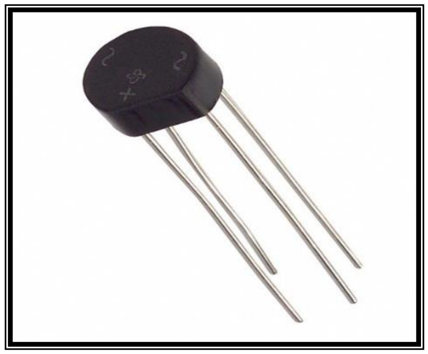

Fig 26: GD W10M Bridge Rectifier

21. Crystal Oscillators:

A crystal oscillator is an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a very precise frequency. This frequency is commonly used to keep track of time (as in quartz wristwatches), to provide a stable clock signal for digital integrated, and to stabilize frequencies for radio transmitters and receivers. The most common type of piezoelectric resonator used is the quartz crystal, so oscillator circuits designed around them became known as "crystal oscillators."

Fig. 27: Crystal Oscillators

22. Pull-up resistor: Pull-up resistors are used in electronic logic circuits to ensure that inputs to logic systems settle at expected logic levels if external devices are disconnected or high-impedance. They may also be used at the interface between two different types of logic devices, possibly operating at different power supply voltages.

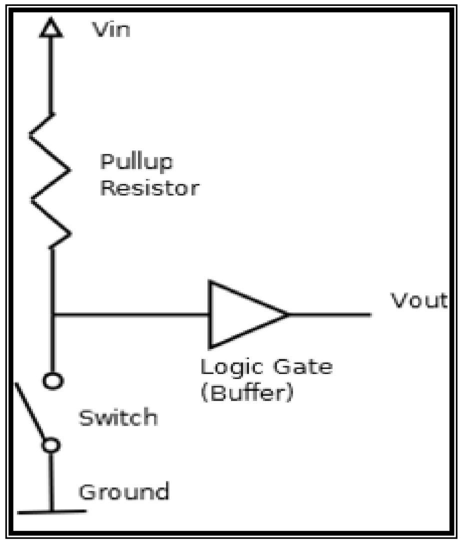

Fig. 28: Pull up resistor

When the switch is open the voltage of the input voltage at the gate goes to ground.
A pull-up resistor weakly "pulls" the voltage of the wire it is connected to towards its voltage source level when the other components on the line are inactive. When all other connections on the line are inactive, they are high-impedance and act like they are disconnected. Since the other components act as though they are disconnected, the circuit acts as though it is disconnected, and the pull-up resistor brings the wire up to the high logic level. When another component on the line goes active, it will override the high logic level set by the pull-up resistor. The pull-up resistor assures that the wire is at a defined logic level even if no active devices are connected to it.
A pull-down resistor works in the same way but is connected to ground. It holds the logic signal near zero volts when no other active device is connected.

23. Ultrasonic Sensor:

The ultrasonic sensor works on the principle of SONAR and RADAR system which is used to determine the distance to an object. An ultrasonic sensor generates the high-frequency sound (ultrasound) waves. An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves.By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object. Sound waves are having specific frequencies or number of oscillations per second. Humans can detect sounds in a frequency range from about 20Hz to 20 KHz. However the frequency range normally employed in ultrasonic detection is 100 KHz to 50MHz. Ultrasonic cleaning uses cavitation bubbles induced by high frequency pressure (sound) waves to agitate a liquid. The agitation produces high forces on contaminants adhering to substrates like metals, plastics, glass, rubber, and ceramics. This action also penetrates blind holes, cracks, and recesses.

Fig 29: Ultrasonic Sensor

Parts of Robotic Arm

Worm Gears:

Worm gears are utilized as a part of applications where vast gear reductions, torque, and braking are important. A portion of the greatest highlights of worm drives is that they accomplish high proportions in a single reduction, and have the capacity to deal with shock loads with less noise.

One preferred standpoint that makes worm drives attractive over other gearboxes is that they have relatively few moving parts in high reduction proportions, essentially two shafts, i.e. input and output, which is backed by four bearings

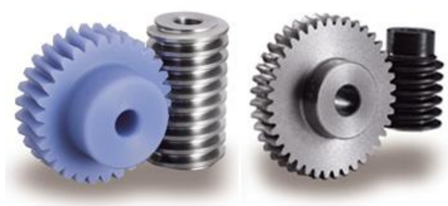

Fig 29: Ultrasonic Sensor

Spur Gears

The most common type of gear is the spur gear. These gears have straight teeth and must be mounted on parallel shafts for their teeth to mesh with those of other gears.

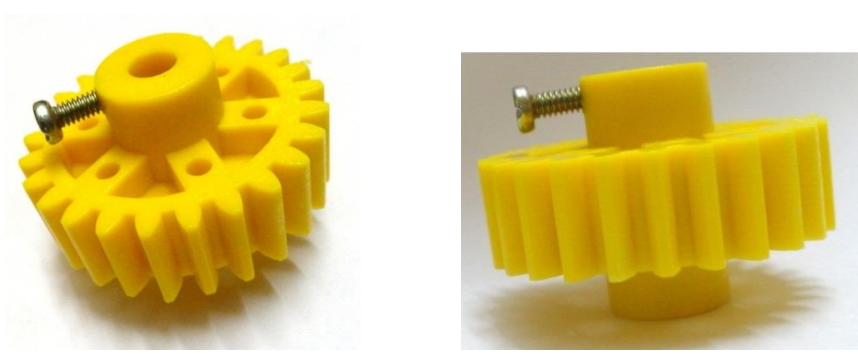

Fig 30: Spur Gears

Chapter 8: Working of Robot:

MATLAB Based GUI Robot works via Bluetooth. It makes a connection with laptop via Bluetooth and using Matlab it will receive serial data where microcontroller decodes the serial data and gives instruction to motor controller to operate. Here we are using GUI technology along with modern robotics.

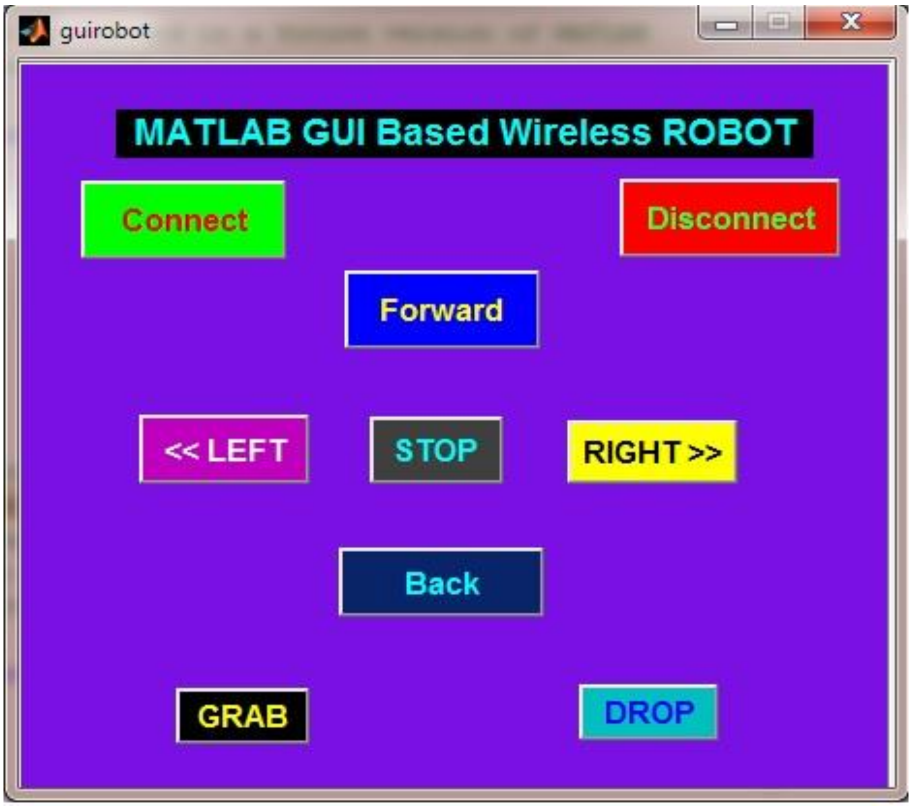

Fig 32: MATLAB GUI for controlling Robot

Step 1: Above fig. shows MATLAB GUI layout for controlling Robot. The Connect push button will establish connection between PC and Robot via Bluetooth. After establishing connection the Robot will ready to take instructions from user.

Step 2: Now user can give instructions to Robot using PC as per choice. For e.g. to move forward, backward, to turn left or right. The Robot can also grab, lift, put and drop the things. When user click the push buttons some data will transmit to the Bluetooth device HC-05 which will further transmit to the microcontroller via serial communication. According to data the Motors will work and Robot will perform specific action.

Step 3: After performing the tasks user can stop the Robot and disconnect the Bluetooth connection.

Block Diagram

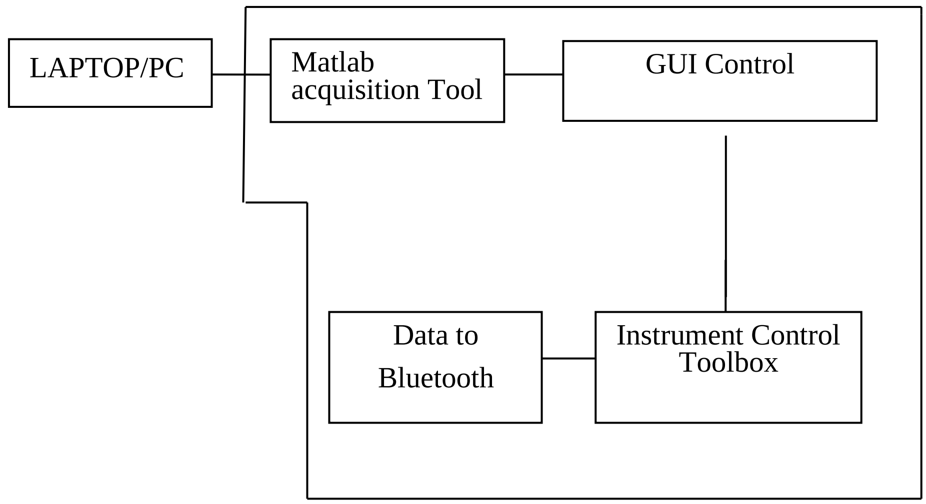

Fig 33: Transmitter section

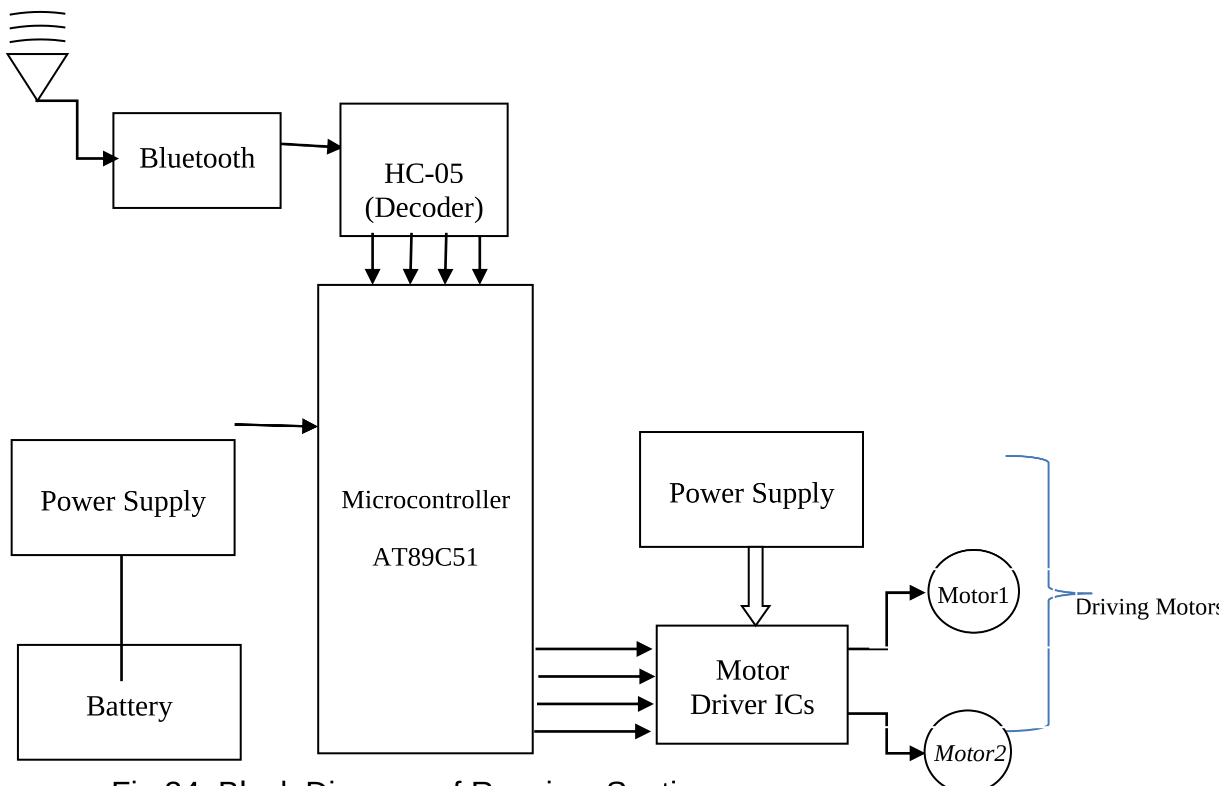

Fig 34: Block Diagram of Receiver Section

Simulation Result

MATLAB simulation:

The code for transmitter section was written and simulated using MATLAB. But the code for receiver section was written in embedded C using Keil. The results were satisfactory we went about with the hardware implementation part. The hardware part was also implemented successfully.

Steps for controlling wireless Robot using MATLAB GUI-

Step 1: Firstly we establish the connection between Robot and laptop via Bluetooth using push button named as "Connect".

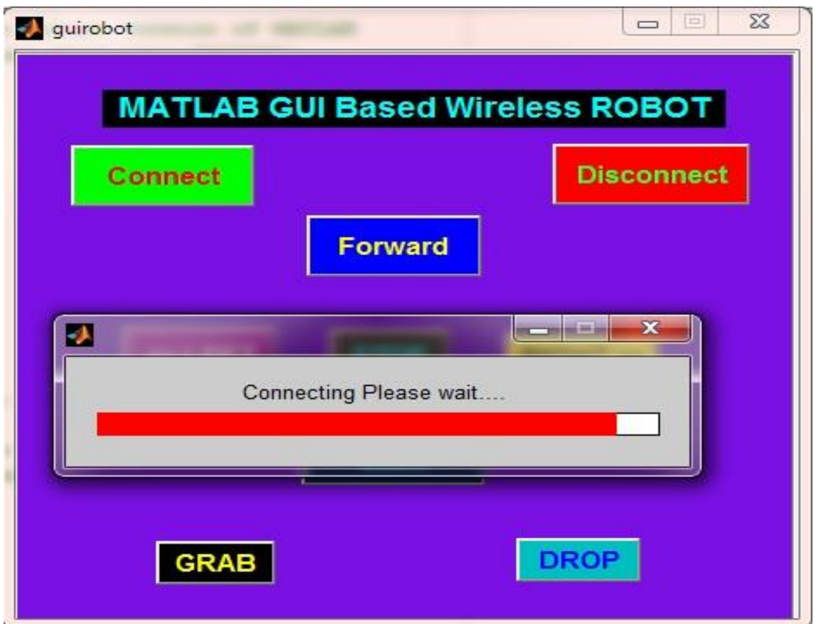

Fig 35: (a) MATLAB simulations

Once the connection established the LED of Bluetooth device would be glowing continuously with the delay of 3 seconds which was glowing continuously without any delay.

Step 2: Now Robot is ready to receive signals. User can give instructions to Robot to move forward and backward, to turn left and right using push buttons.

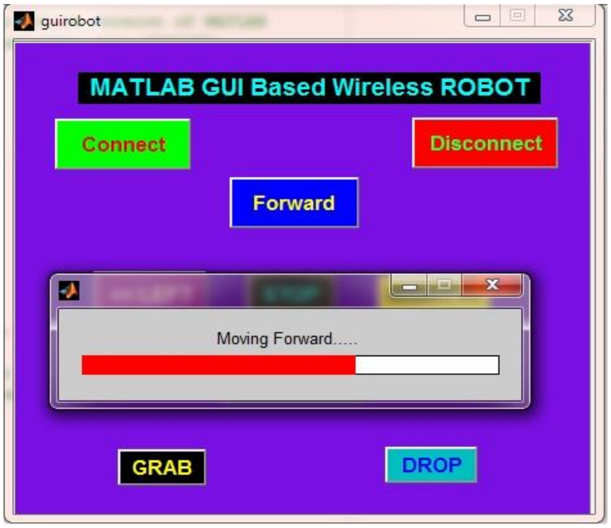

Fig 35: (b) MATLAB simulations

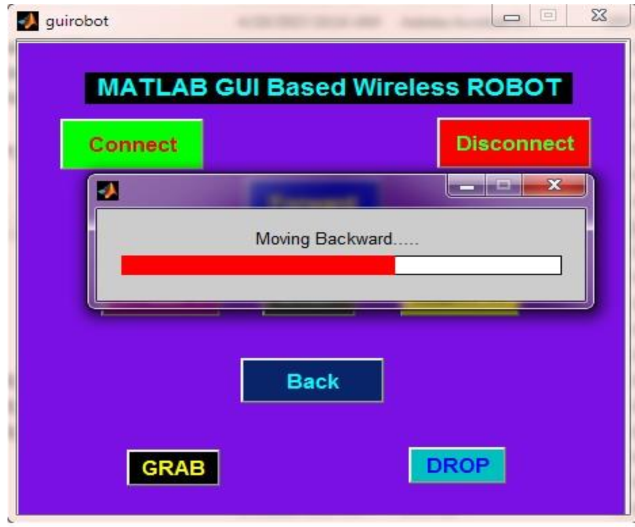

Fig 35: (c) MATLAB simulations

Step 3: After Robot has completed its task user can stop the Robot using push button named as “STOP”. And also user can disconnect the Robot using push button named as “Disconnect”.

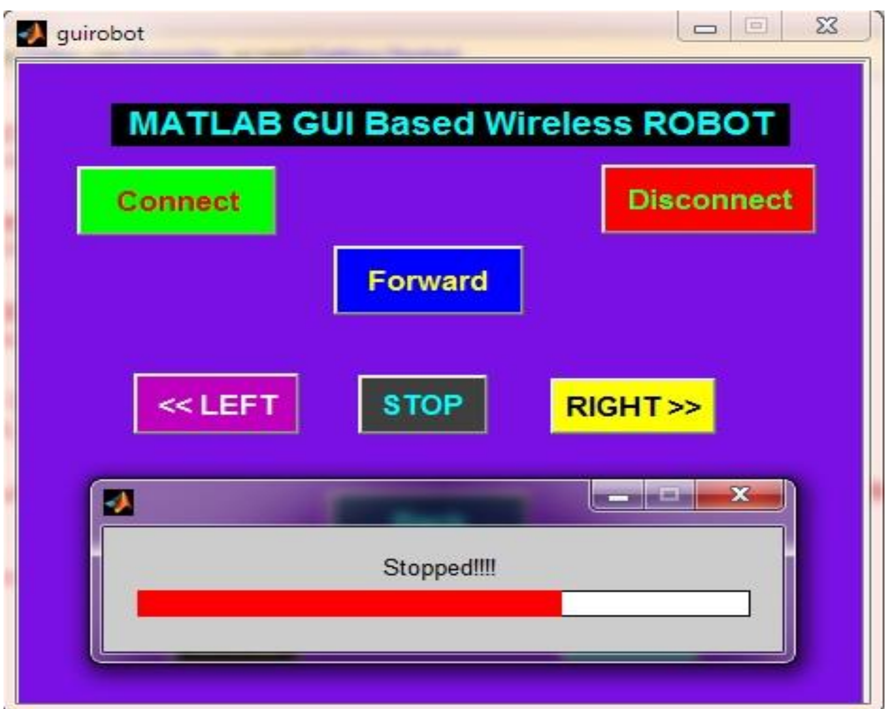

Fig 35: (d) MATLAB simulations

Chapter 9: Embedded C

Introduction to Embedded C

Embedded C is a set of language extensions for the C Programming language by the C. In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing. Embedded C uses most of the syntax and semantics of standard C, e.g., main () function, variable definition, data type declaration, conditional statements (if, switch, case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.

During infancy years of microprocessor based systems, programs were developed using assemblers and fused into the EPROMs. There used to be no mechanism to find what the program was doing. LEDs, switches, etc. were used to check for correct execution of the program. Some ‘very fortunate’ developers had In-circuit Simulators (ICEs), but they were too costly and were not quite reliable as well. As time progressed, use of microprocessor-specific assembly-only as the programming language reduced and embedded systems moved onto C as the embedded programming language of choice.

C is the most widely used programming language for embedded processors/ controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements. As assembly language programs are specific to a processor, assembly language didn’t offer portability across systems. To overcome this disadvantage, several high level languages, including C, came up. Some other languages like PLM, Modula-2, Pascal, etc. also came but couldn’t find wide acceptance. Amongst those, C got wide acceptance for not only embedded systems, but also for desktop applications.

Advantages:

- It is small and simpler to learn, understand, program and debug.
- Compared to assembly language, C code written is more reliable and scalable, more portable between different platforms.
- C compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.
- It is fairly efficient.
- It supports access to I/O and provides ease of management of large embedded projects.

Java is also used in many embedded systems but Java programs require the Java Virtual Machine (JVM), which consumes a lot of resources. Hence it is not used for smaller embedded devices.

Chapter 10: Embedded C code

```
void setup()
{

  Serial.begin(9600);
  pinMode(5, OUTPUT);//M+
  pinMode(6, OUTPUT);//M-
  pinMode(7, OUTPUT);//M2+
  pinMode(8, OUTPUT);//M2-
  pinMode(9, OUTPUT);//M3+
  pinMode(10, OUTPUT);//M3-
  pinMode(11, OUTPUT);//M3+
  pinMode(12, OUTPUT);//M3-
}
void Forward()
{
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
   digitalWrite(7, HIGH);
  digitalWrite(8, LOW);
}
void Back()
{
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
   digitalWrite(7, LOW);
  digitalWrite(8, HIGH);
}

void Left()
{

  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
   digitalWrite(7, LOW);
  digitalWrite(8, HIGH);
}
void Right()
{
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
   digitalWrite(7, HIGH);
  digitalWrite(8, LOW);
```

```
}


void Stop()
{
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
   digitalWrite(7, LOW);
  digitalWrite(8, LOW);
}

void armr()
{
  digitalWrite(9, LOW);
  digitalWrite(10, HIGH);

}

void arml()
{
  digitalWrite(9, HIGH);
  digitalWrite(10, LOW);

}

void gripopen()
{
  digitalWrite(11, HIGH);
  digitalWrite(12, LOW);

}

void gripclose()
{
  digitalWrite(11, LOW);
  digitalWrite(12, HIGH);

}

void gripstop()
{
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(9, LOW);
  digitalWrite(10, LOW);
}
```

```
void loop()
{


  if(Serial.available())
  {
    switch(Serial.read())// Recieve data from bluetooth
    {
      case 'f':
        Forward();
        break;
      case 'b':
        Back();
        break;
      case 'l':
        Left();
        break;
      case 'r':
        Right();
        break;
      case 's':
        Stop();
        gripstop();
        break;
      case 'm':
        armr();
        break;
      case 'n':
        arml();
        break;
      case 'a':
        gripopen();
        break;
      case 'c':
        gripclose();
        break;
      case 'x':
       gripstop();
        break;
    }
  }
}
```

Chapter 11: Advantages

- Replace soldiers in dangerous missions, such as crawling through caves or in street-to-street urban combat, reducing casualties.
- Reduce civilian casualties if used properly and if sufficient ethical programming could be developed.
- Act as a “force multiplier.” One human fighter could command a squad of robots working semi autonomously.
- Make faster decisions than humans, an important advantage on the modern battlefield.
- Be unaffected by anger, revenge, hunger, fear, fatigue, or stress.
- Use video or other sensors to monitor human soldiers on both sides of a battle for violations of the laws of war.
- Refuse to carry out an unethical or illegal commands, something a human soldier might be pressured not to do.

Chapter 12: Future Scope and Applications

Further advancements can be done by decreasing the time delay in voice mode and sensors can be attached to the wheelchair to avoid collision. Another further advancement in can be done is headset can be apply in the system. When the headsets are attached to patients head and if he/she want to do movement then just move the head and chair move according to that. And we can also add the feature of Accelerometer for those people who cannot handle robot movement by their hand. We can mount mobile on the head of paraplegic person and according to the movement of the head of that person the robot will move.

There are many unsolved problems and fundamental challenges for robotics at a very high level manipulation and physical interaction with the real world. We need concerted modeling and control efforts together with the development of good hardware to make arms and hands that can perform anything but the simplest of pick-and-place operations that are prevalent in industry. The pick and place robot is having the very vast area of applications.

The future scope applications of this project are:

1**.** Bomb diffusion: After few modifications of the pick and place mechanism we can improve the robot for the bomb diffusion purpose. Using the web cam we can train the robot to diffuse the bomb hence without putting human life in danger we can fight against the terrorism.

2. Our robot can handle dangerous chemicals in chemical lab our in nuclear reactor labs which are hazardous to human body.

3. We can use webcam in this robot to perform many tasks that human cannot or dangerous for human to handle.

4. With some modifications this robot can be used for helping the physically challenged people

Chapter 13: Conclusion

This project is a simple demonstration of the wireless robot that we wish to control using microcontrollers. In this project we have demonstrated to link the MATLAB to control microcontroller output. Here only pushbuttons are to be used by the user so no need to write the program again and again once the declaration has been done.

This project doesn't need exceptional coding skills for microcontroller and MATLAB. A little knowledge of both can help the user fulfil his basic home and industrial tasks.

Chapter 14: References:

[1] O. González, M. Rodríguez, A. Ayala, J. Hernández & S. Rodríguez, “Application of PIC and microcontrollers in the measurement and control of parameters in industry” in International Journal of Electrical Engineering Education, 41/3, 2004, pp. 266-274.

[2] V. Nagaphanindra and B. Suresh Ram, “Wireless Remote Control Car Based on ARM7” in International Journal of Engineering Trends and Technology (IJETT) Volume 5 Number 5 - Nov 2013.

[3] M. L. Glaze, “The Design and Implementation of a GUI-Based Control Allocation Toolbox in the MATLAB® Environment”, Msc. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, 85 p. 2008.

Websites:

[4] www.google.co.in

[5] www.engineersgarage.com

[6] www.mathworks.com

[7] www.wikipedia.org

[8] www.robokits.com

[9] www.TheEngineeringProjects.com