

ROBOTIC ARM

¹Miss.Sayali V. Vanjari, ²Miss.Poonam Ghewande, ³Miss.Kalyani Deshmukh, ⁴Miss. Rasika Sawale
^{1 2 3 4}BE CSE Scholars

Department of Computer Science and Engineering, Anuradha Engineering College Chikhli, India

Abstract : In today's world there is an increasing need to create artificial arms for different in human situations where human interaction is difficult or impossible. They may involve taking readings from an active volcano to diffusing a bomb. Here we propose to build a robotic arm controlled by natural human arm movements. A robotic arm is a robotic manipulator, usually programmable, with similar functions to a human arm. Humans pick things up without thinking about the steps involved. In order for a robot or a robotic arm to pick up or move something, someone has to tell it to perform several actions in a particular order from moving the arm, to rotating the "wrist" to opening and closing the "hand" or "fingers". So, we can control each joint. . This project presents a three joint automatic robotic arm which can be used in industries to do repetitive task such as moving the things from one place to another place. The robotic arm controlled method is accepted to overcome the problem such as placing or picking object that away from the user, pick and place object in very fast and easy manner.

Keywords: End effector, Shaft, Degrees of freedom.

CHAPTER 1

Introduction

Nowadays, robots are increasingly being integrated into working tasks to replace humans especially to perform the repetitive task. In general, robotics can be divided into two areas, industrial and service robotics. International Federation of Robotics (IFR) defines a service robot as a robot which operates semi- or fully autonomously to perform services useful to the wellbeing of humans and equipment, excluding manufacturing operations. These robots are currently used in many fields of applications including office, military tasks, hospital operations, dangerous environment and agriculture. Besides, it might be difficult or dangerous for humans to do some specific tasks like picking up explosive chemicals, defusing bombs or in worst case scenario to pick and place the bomb somewhere for containment and for repeated pick and place action in industries. Therefore a robot can be replaced human to do work.[1]

Robots are increasingly being integrated into working tasks to replace humans especially to perform the repetitive task. In general, robotics can be divided into two areas, industrial and service robotics. International Federation of Robotics (IFR) defines a service robot as a robot which operates semi- or fully autonomously to perform services useful to the well-being of humans and equipment, excluding manufacturing operations .

On the other hand, internet nowadays is becoming the center for everything. People tend to get online rather than doing household chores. Internet is now everywhere, compare to the last decades where internet is only wired, and people needs to be in front of the computer to access the internet but nowadays, internet is just at the tip of your finger. This is an advantage to introduce robot to household. The robot body was prepared mechanically and electrical components were chosen to be suitable to be used as a robotic arm. [1]

The robot is controlled using Arduino Uno as the brain of the robot, connected to the internet via Arduino Ethernet Shield as the interface for Arduino Uno to the internet. Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. The movement of the robot is controlled by inputting the desired degree of movement of the robotic arm and then the robotic arm will move to the desired movement that has been inputted. There is also a pre-programmed movement of the robotic arm with a click of a button. [1]

1.2 Robotic arm definition:-

A robotic arm is a robot manipulator, usually programmable, with similar functions to a human arm. The links of such a manipulator are connected by joints allowing either rotational motion (such as in an articulated robot) or translational (linear) displacement. The links of the manipulator can be considered to form a kinematic chain. The business end of the kinematic chain of the manipulator is called the end effectors and it is analogous to the human hand. The end effectors can be designed to perform any desired task such as welding, gripping, spinning etc., depending on the application. The robot arms can be autonomous or controlled manually and can be used to perform a variety of tasks with great accuracy. [5]

The robotic arm can be fixed or mobile (i.e. wheeled) and can be designed for industrial or home applications. This report deals with a robotic arm whose objective is to imitate the movements of a human arm using accelerometers as sensors for the data acquisition of the natural arm movements. This method of control allows greater flexibility in controlling the robotic arm rather than using a controller where each actuator is controlled separately. The processing unit takes care of each actuator's control signal according to the inputs from accelerometer, in order to replicate the movements of the human arm. Figure 1 shows the block diagram representation of the system to be designed and implemented.[5]

The word robot originated from the Czech word robota, meaning work. A definition used by the robot institute of America is: "A robot is a programmable multifunction manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks". In this highly developing society, time and man power are critical constrains for completion of task. The automation is playing important role to save human efforts in most of the regular and frequently carried work. The idea that machines can begin to imitate human actions, even in ways we have not thought of, the main motives for the creation of robot have been very practical. First, as modern industry has become more complex, there has been a growing need for getting work done in environments that are dangerous for humans. As an example, work in a nuclear reactor plant often requires contact with radioactive materials. Second, as robots became more advanced and less expensive, they are being set up in industry situations where working conditions are not so much dangerous as unpleasant for various reasons. These situations typically involve high degrees of the following: - Heat, Noise, Poisonous gases, Risk of injury by machines, Monotonous, boring work. Robots have already taken over a number of such unpleasant jobs in industry- welding in automobile factories, which involves heat, noise and heavy exertion. Robots are obedient, untiring and precision welders. Simple robots do

many routine jobs in industry. Pick and place robots are useful in simple assembly operations such as stuffing printed circuit boards and loading and unloading parts from machines[5]

1.3 PROPOSED MODEL:

BLOCK DIAGRAM

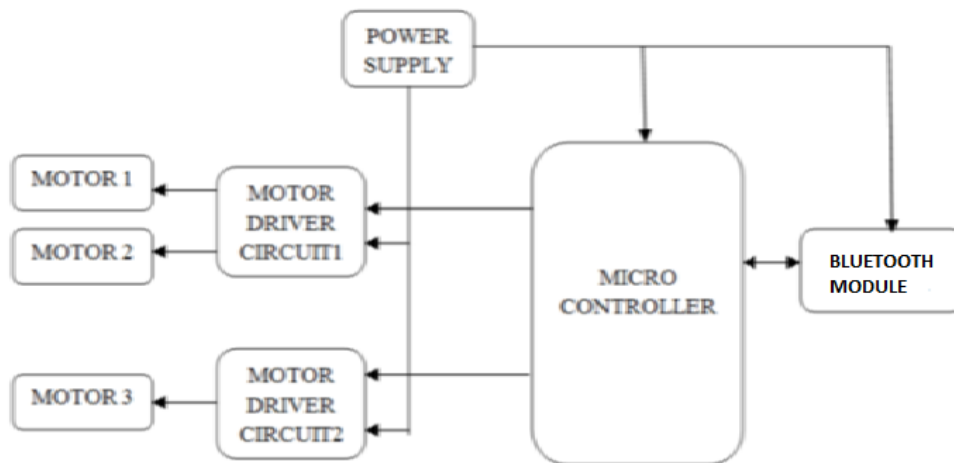


Fig 1 : Block Diagram

A. MICROCONTROLLER :-

A microcontroller is a high integrated functional computer system-on-a-chip. It contains an integrated memory and programmable input/output peripherals. Microcontrollers often operate at very low speed. They consume relatively little power. It is used to control the motor activation and deactivation operations and also reads sensor signals.

B. MOTOR DRIVER CIRCUIT:-

In order to provide the required amperage to the motor using the low current signal from the microcontroller, motor driver circuit is used. A motor controller is a device or group of devices that serves to govern in some predetermined manner the performance of an electric motor. A motor controller might include a manual or automatic means for starting and stopping the motor, selecting forward or reverse rotation, selecting and regulating the speed, regulating or limiting the torque, and protecting against overloads and faults.

C. MOTORS :-

An electric motor is an electromechanical device that converts electrical energy into mechanical energy. Electric motors can be powered by direct current sources, such as from batteries. Microcontrollers command these motors through the driver circuit to take the necessary action.

D. POWER SUPPLY :-

A regulated power supply is an embedded circuit, the function of which is to supply a stable voltage, to a circuit or device that must be operated within certain power supply limits. This is used to supply the power to the microcontroller and the driver circuits.

2. Literature Review:-

There are various ways in which a robotic arm may be controlled. In the past there have been many researchers working to control robotic arm through computer terminals, Joysticks, even interfacing them with the internet so they can be controlled from anywhere in the world. Usually most of the robotic arms are controlled by a central controller which makes use of values taken in from the terminal that are entered by the user at the terminal to move the arm to a particular coordinate in space. This makes the control very difficult as the control values of the motors are very difficult to predict to achieve a particular movement. This is easily achieved by our project.

This Project represents a simple accelerometer controlled robotic arm using Atmega328 powered embedded system as the core of this robot and also a Computer to interface the robot with the bluetooth. The robot does not require training because the robotic arm is fully controlled by the user. This interfacing is done using wired communication but it can easily be switched to wireless with ease.

✓ 2.1 Early Modern Robots and Robotic Arms:

Now, with the advent of machines, electronics, automation and the incorporation of solid-state transistors instead of vacuum tubes, the evolution of the microcircuit and more rapid computer systems, the stage was set for early modern robotic arm evolution. The first "position controlling apparatus" was patented in 1938 by Willard Pollard. This was a spray finishing robotic arm that had five degrees-of-freedom and an electrical control system. Although Pollard never built his arm, his design and interest in a mechanized application for automated robotic arms would spur on the ingenuity of others. Harold A. Roselund, working for De Vilbiss, developed another sprayer that was indeed manufactured. Both arms were useful and task oriented for their time and every movement at the respective joints in unique ways.

✓ 2.2 "The Multiple-Function Intelligent Robotic Arms"

The main design goal of this robotic arm was to present the following functions: Chinese calligraphy, fancy dancing, weightlifting and colour classification. Another design goal was to minimize cost and maximize performance.[7]

Hence it is designed in a way so as to perform many tasks as a human arm. The simplicity of its design gave it the way to perform motion as good as a human arm with improved values of strength and speed.[7]

✓ 2.3 "Development of a Robotic-Arm Controller by using Hand Gesture Recognition"

This robotic arm controller is using image processing in the field of Human-Machine Interaction. There are two methods used to control the robotic arm, the main task of them is to get the hand gesture data without using tool that helps the system to extract data easier (ex. glove or wrist band). The first method is comparing of all pre-stored data in the database at the Template Matching Algorithm, the second method is Signature Signal, distance signal between edge of the hand and centre of hand. These necessarily use the gyro-sensors as the computation of change in angular values shall be obtained at time of operation.[5]

The subsequent operations are decided based on these changes.[5]

✓ 2.4 “The Development of Six D.O.F. Robot Arm for Intelligent Robot”

The aim of this paper is to develop a six D.O.F robot arm for an intelligent robot. As for the base or shoulder part, the harmonic responding drive, which has some advantages, was used to ensure the weight of the whole arm could be supported. This provides a degree of freedom along with supporting the assembly. The kinematic equations of the system have been studied and shown in this. The degrees in which the robotic unit make moves or acquires motion necessarily contribute to improvement of operation. added or updated motion represents ease of operation in certain operating context.[6] At the end, we investigate some experiments; the computer system based controller can command the robot arm to pick a box through a moving path successfully.[6]

3. Project Overview:-

In this Project, the hardware and software function are combined to make the system reliable.

The ATmega328 will be interfacing the robot with the Bluetooth,L2938D IC, Motor Driver and Android Mobile i.e. stepper motors which will control the movement of the robot respectively.

The chapter that follows describe the hardware (Chapter 2), which is followed by the description of the software being used (Chapter 3) Chapter 4 describes the implementation of the project and Chapter 5 future scope and Chapter 6 conclusion of project.

CHAPTER 2 Hardware Design And Description

This chapter describes the hardware that is being used in the project.

2.1 Hardware Requirements

- 1) ATmega328 arduino uno
- 2) L293D Ic Motor Driver
- 3) bluetooth HC-05
- 4) stepper motor 3
- 5) android mobile

1)AT mega328 Arduino uno:-

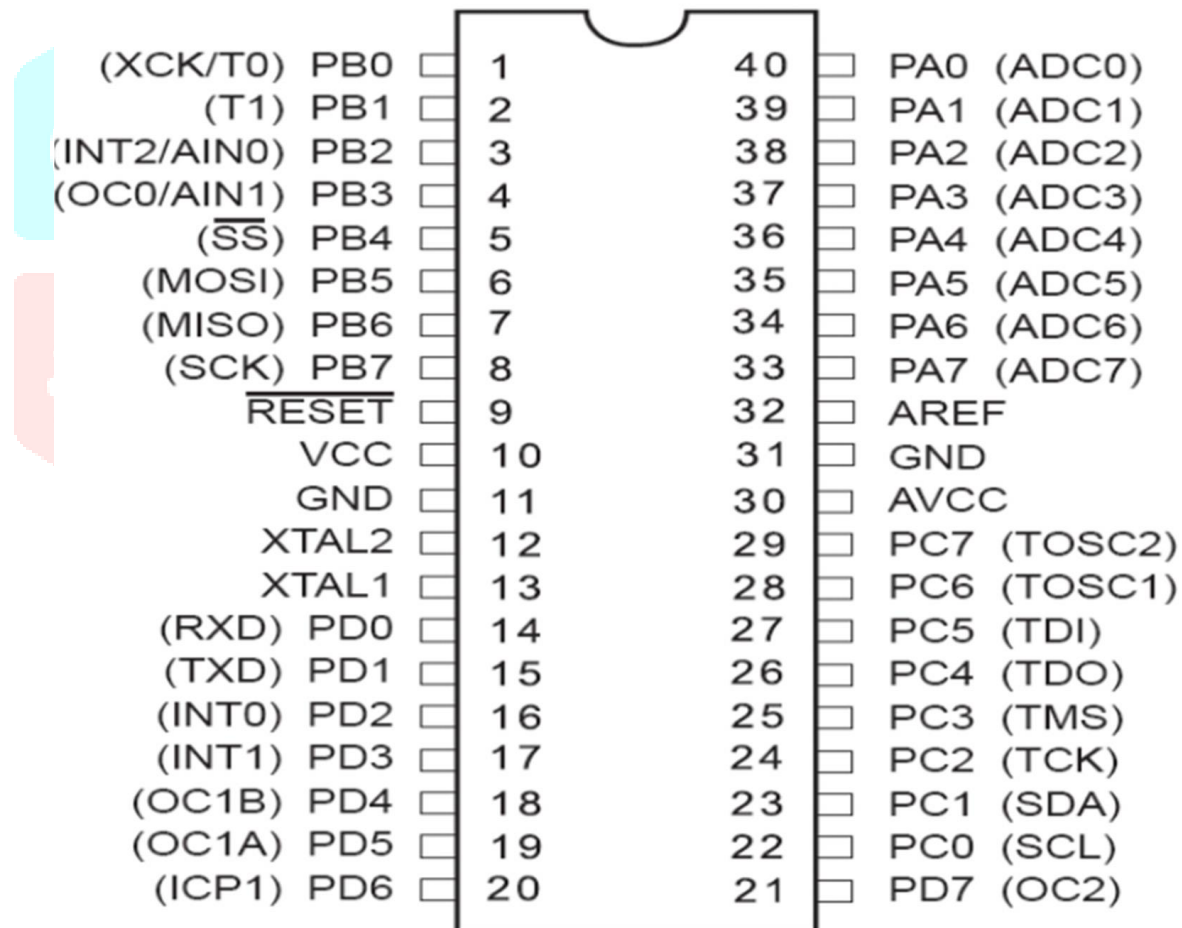


Fig 2:- Pin Configuration of ATmega328

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 a 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. [4]

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.. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0.[4]

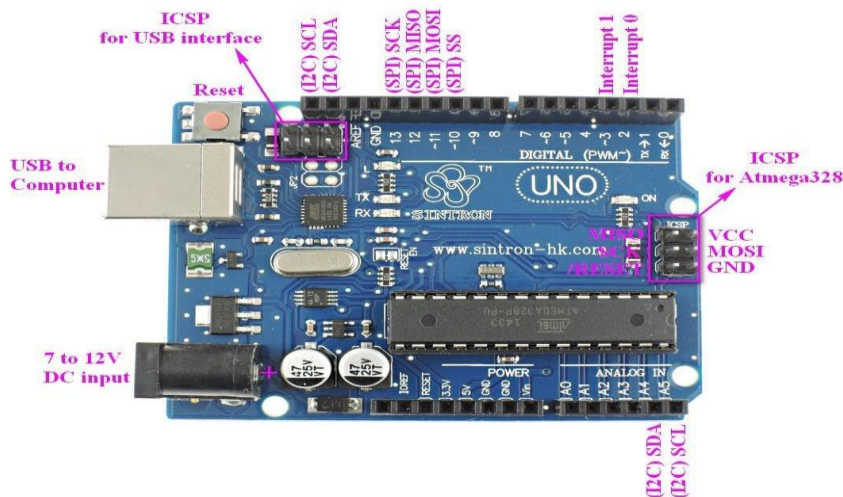


Fig 3:- AT mega328arduino Uno

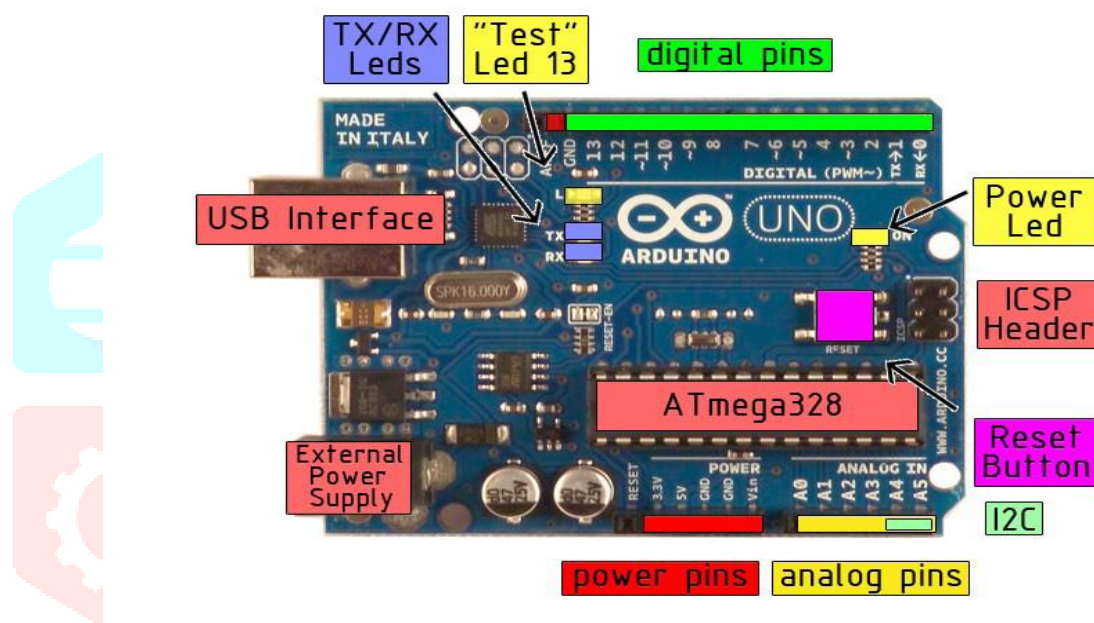


Fig 4:- AT mega328arduino Uno board

➤ Technical Specification:-

- ✓ Microcontroller ATmega328
- ✓ Operating Voltage 5V
- ✓ Input Voltage (recommended) 7-12V
- ✓ Input Voltage (limits) 6-20V
- ✓ Digital I/O Pins 14 (of which 6 provide PWM output)
- ✓ Analog Input Pins 6
- ✓ DC Current per I/O Pin 40 mA
- ✓ DC Current for 3.3V Pin 50 mA
- ✓ Flash Memory 32 KB of which 0.5 KB used by bootloader
- ✓ SRAM 2 KB
- ✓ EEPROM 1 KB
- ✓ Clock Speed 16 MHz

➤ Power:-

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.[4]

➤ The power pins are as follows:

- ✓ **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 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.
- ✓ **3V3.** 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 bootloader); 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 pinMode(), digitalWrite(), and digitalRead() 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 kOhms. In addition, some pins have specialized functions:

- 1) **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 .
- 2) **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.
- 3) **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analog Write() function.
- 4) **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.
- 5) **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, 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:
- 6) **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.
- ✓ **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 *.inf file is required.[4]

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; see the To use the SPI communication.[6]

➤ **Programming:-**

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/ATmega328" from the Tools > Board menu (according to the microcontroller on your board). [4]

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also by pass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header.[8]

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer(overwriting the DFU bootloader). Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. [8]

The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.[4]

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.[4]

➤ **USB Overcurrent Protection:-**

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.[8]

➤ **Physical characteristics:-**

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.[7]

➤ **How to use Arduino?**

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the

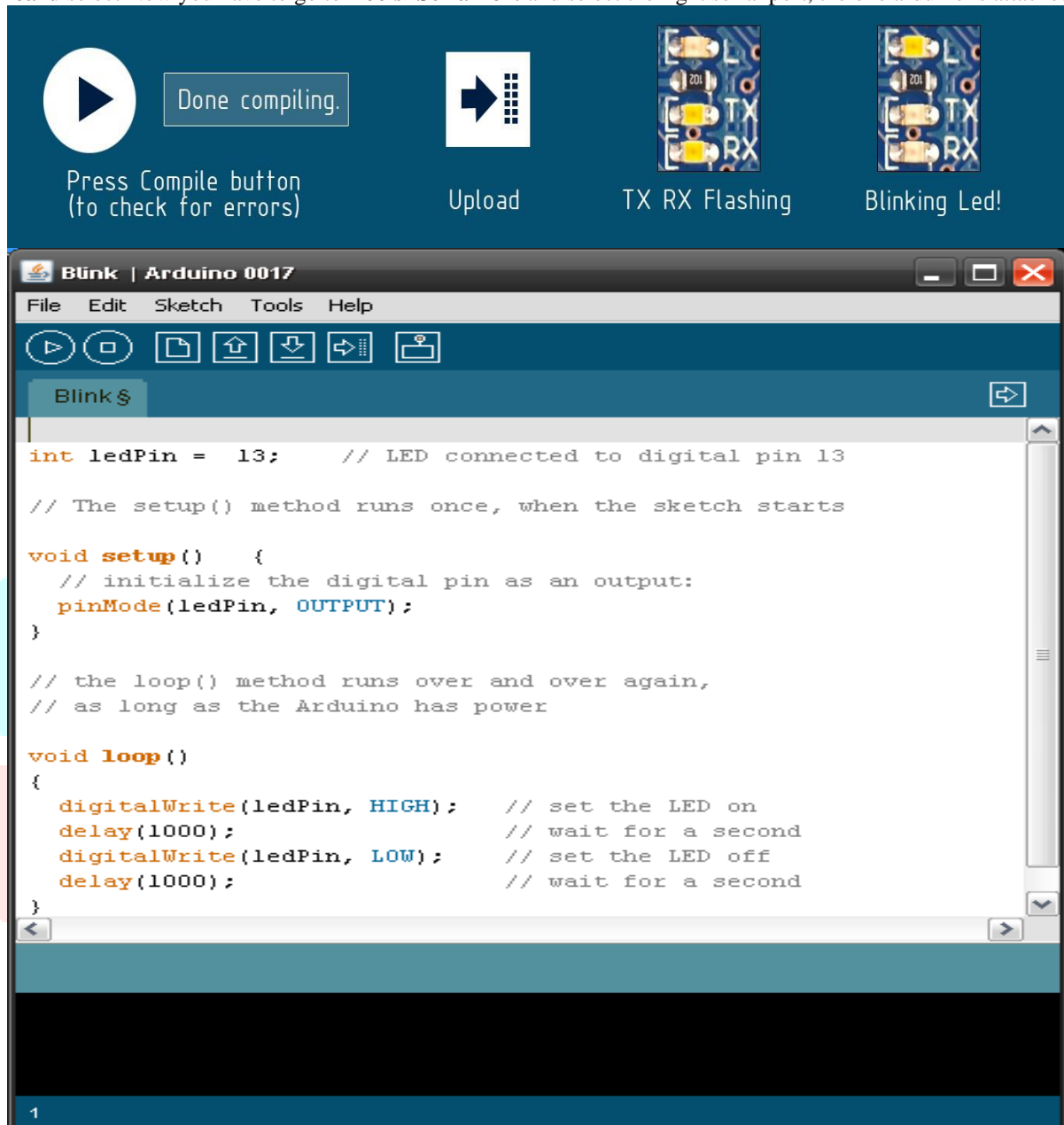
Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP). Arduino is a cross-platform program

➤ Blink led:-

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "helloworld", select

File>Sketchbook>Arduino-0017>Examples>Digital>Blink

In **Tools>Board** select Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.



➤Terms and condition:-

1.Warranties:-

- i. The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.[8]
- ii. If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.[4]
- iii. Except as set forth above ,products are provided as is and with all faults. The Producer disclaims all other warranties, express or implied, regarding products, including but not limited to,any implied warranties of merchantability or fitness for a particular purpose[1].
- iv. Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

- v. The Arduino[®] products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino[®] products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.[4]
- vi. Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.[4]

2. Indemnification:-

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.[8]

3. Consequential Damages Waiver:-

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.[4]

4. Changes to specifications:-

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.[4]

2) L293D Ic Motor Driver:-

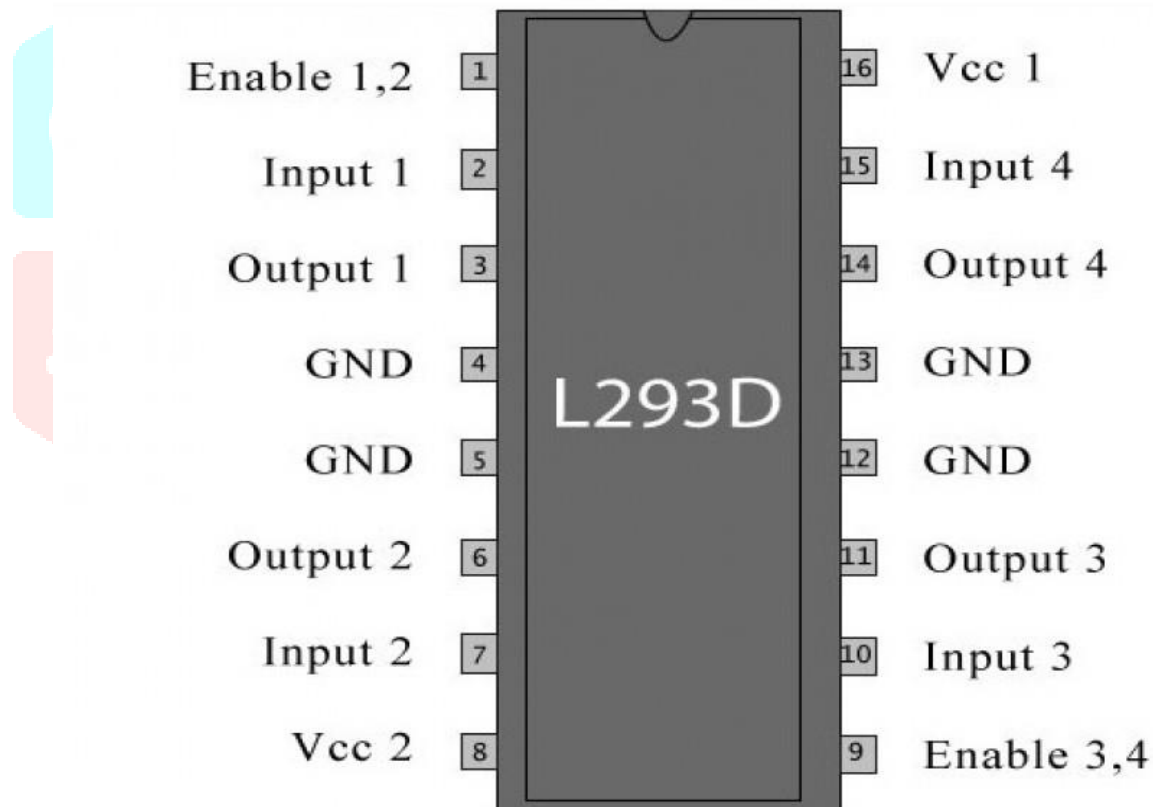


Fig 5 :- L293D Ic Motor Driver

➤ Introduction:-

The L293D is a quadruple half H-bridge bidirectional motor driver IC that can drive current of up to 600mA with voltage range of 4.5 to 36 volts. It is suitable to drive small DC-Geared motors, bipolar stepper motor etc. L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC. In a single L293D chip there are two h-Bridge circuit inside the IC which can rotate two dc motor independently. H-bridge is a circuit which allows the voltage to be flown in either direction. H-bridge IC are ideal for driving a DC motor. Due its size it is very much used in robotic application for controlling DC motors.[6]

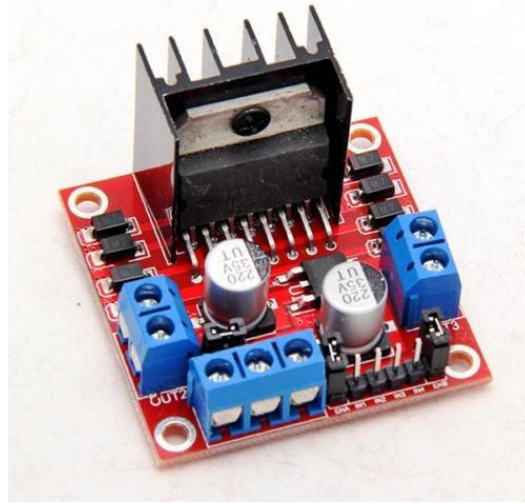


Fig 6 :-Motor driver

➤ Working of L293D Ic:-

There are 4 input pins for L293d, pin 2,7 on the left and pin 15 ,10 on the right as shown on the pin diagram. Left input pins will regulate the rotation of motor connected across left side and right input for motor on the right hand side. The motors are rotated on the basis of the inputs provided across the input pins as LOGIC 0 or LOGIC 1. For rotating the motor in clockwise direction the input pins has to be provided with Logic 1 and Logic 0.Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.[6]

➤ Specifications:-

- ✓ Supply Voltage Range 4.5V to 36V
- ✓ 600-mA Output current capability per driver
- ✓ Separate Input-logic supply
- ✓ It can drive small DC-g geared motors, bipolar stepper motor.
- ✓ Pulsed Current 1.2-A Per Driver
- ✓ Thermal Shutdown
- ✓ Internal ESD Protection
- ✓ High-Noise-Immunity Inputs

➤ Applications:-

- DC and stepper motor drives
- Position and velocity servomechanisms

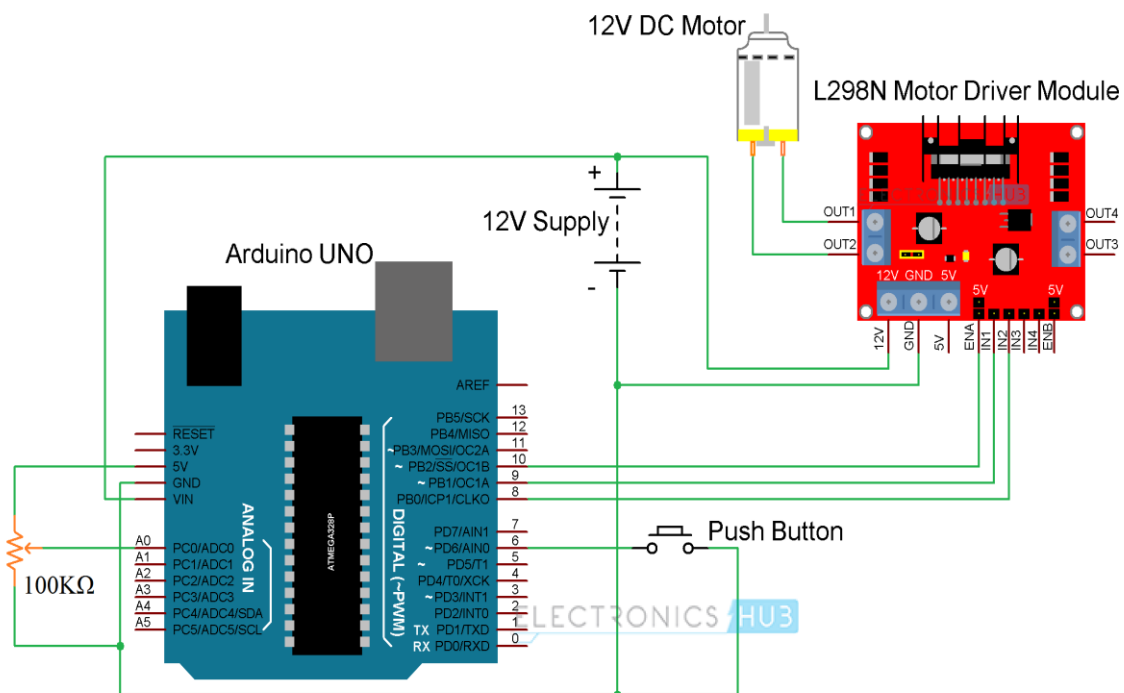


Fig 7 :-connection of L293D Ic Motor Driver

3) Bluetooth HC-05:-

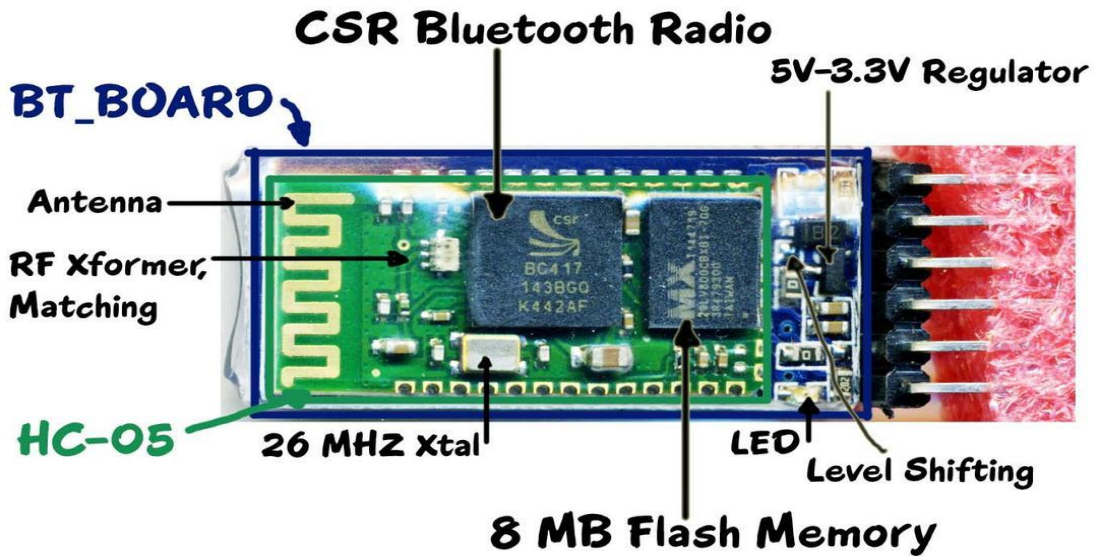


Fig 8:- Bluetooth HC-05

➤ Introduction:-

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature). [5]

➤ Bluetooth module HC-05:-

The Bluetooth module HC-05 is a MASTER/SLAVE module. By default the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections. Master module can initiate a connection to other devices. The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project, etc. [6]

➤ Hardware Features:-

- ✓ Typical -80dBm sensitivity.
- ✓ Up to +4dBm RF transmits power.
- ✓ 3.3 to 5 V I/O.
- ✓ PIO(Programmable Input/Output) control.
- ✓ UART interface with programmable baud rate.
- ✓ With integrated antenna.
- ✓ With edge connector.

➤ Software Features:-

- ✓ Slave default Baud rate: 9600, Data bits:8, Stop bit:1,Parity:No parity.
- ✓ Auto-connect to the last device on power as default.
- ✓ Permit pairing device to connect as default.
- ✓ Auto-pairing PINCODE:”1234” as default.

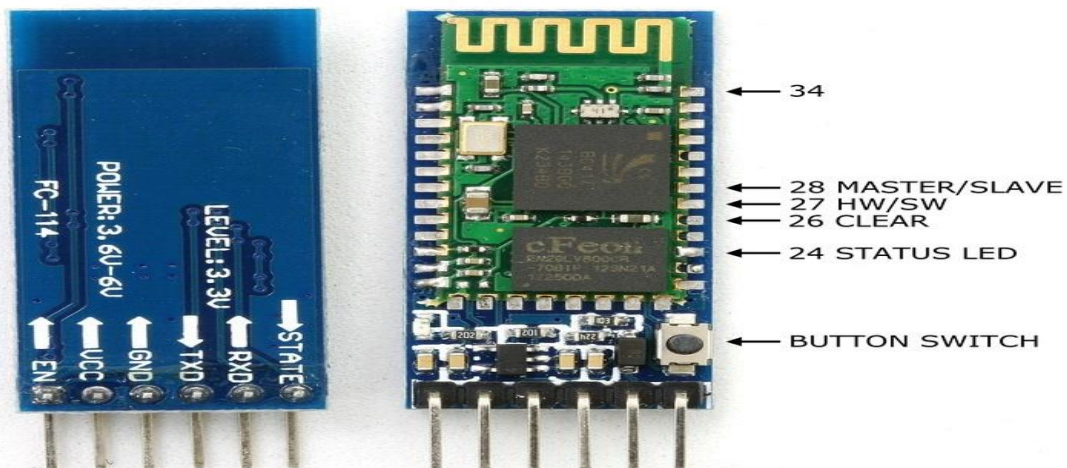


Fig 9:- Pin Description

➤ Pin Description:-

The HC-05 Bluetooth Module has 6 pins. They are as follows:

1) ENABLE:-

When enable is pulled LOW, the module is disabled which means the module will not turn on and it fails to communicate. When enable is left open or connected to 3.3V, the module is enabled i.e the module remains on and communication also takes place.

2) Vcc:-

Supply Voltage 3.3V to 5V

3) GND:-

Ground pin

4) TXD & RXD:-

These two pins acts as an UART interface for communication.

5) STATE:-

It acts as a status indicator. When the module is not connected to / paired with any other Bluetooth device, signal goes Low. At this low state, the led flashes continuously which denotes that the module is not paired with other device. When this module is connected to/paired with any other bluetooth device, the signal goes High. At this high state, the led blinks with a constant delay say for example 2s delay which indicates that the module is paired.

6) BUTTON SWITCH:-

This is used to switch the module into AT command mode. To enable AT command mode, press the button switch for a second. With the help of AT commands, the user can change the parameters of this module but only when the module is not paired with any other BT device. If the module is connected to any other bluetooth device, it starts to communicate with that device and fails to work in AT command mode.

✚ How to connect HC05 Bluetooth module with Arduino Uno?

✓ Hardware and Software Required

- HC-05 Bluetooth Module
- Arduino Uno
- Arduino IDE(1.0.6V)

➤ Hardware Connections:-

As we know that Vcc and Gnd of the module goes to Vcc and Gnd of Arduino. The TXD pin goes to RXD pin of Arduino and RXD pin goes to TXD pin of Arduino i.e(digital pin 0 and 1). The user can use the on board Led. But here, Led is connected to digital pin 12 externally for betterment of the process.[6]

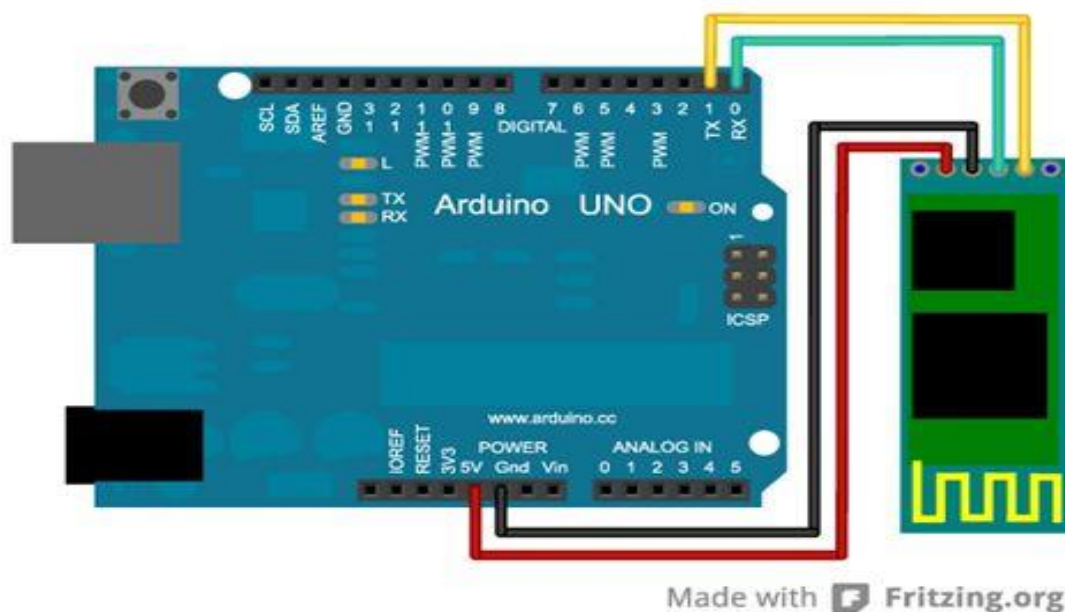


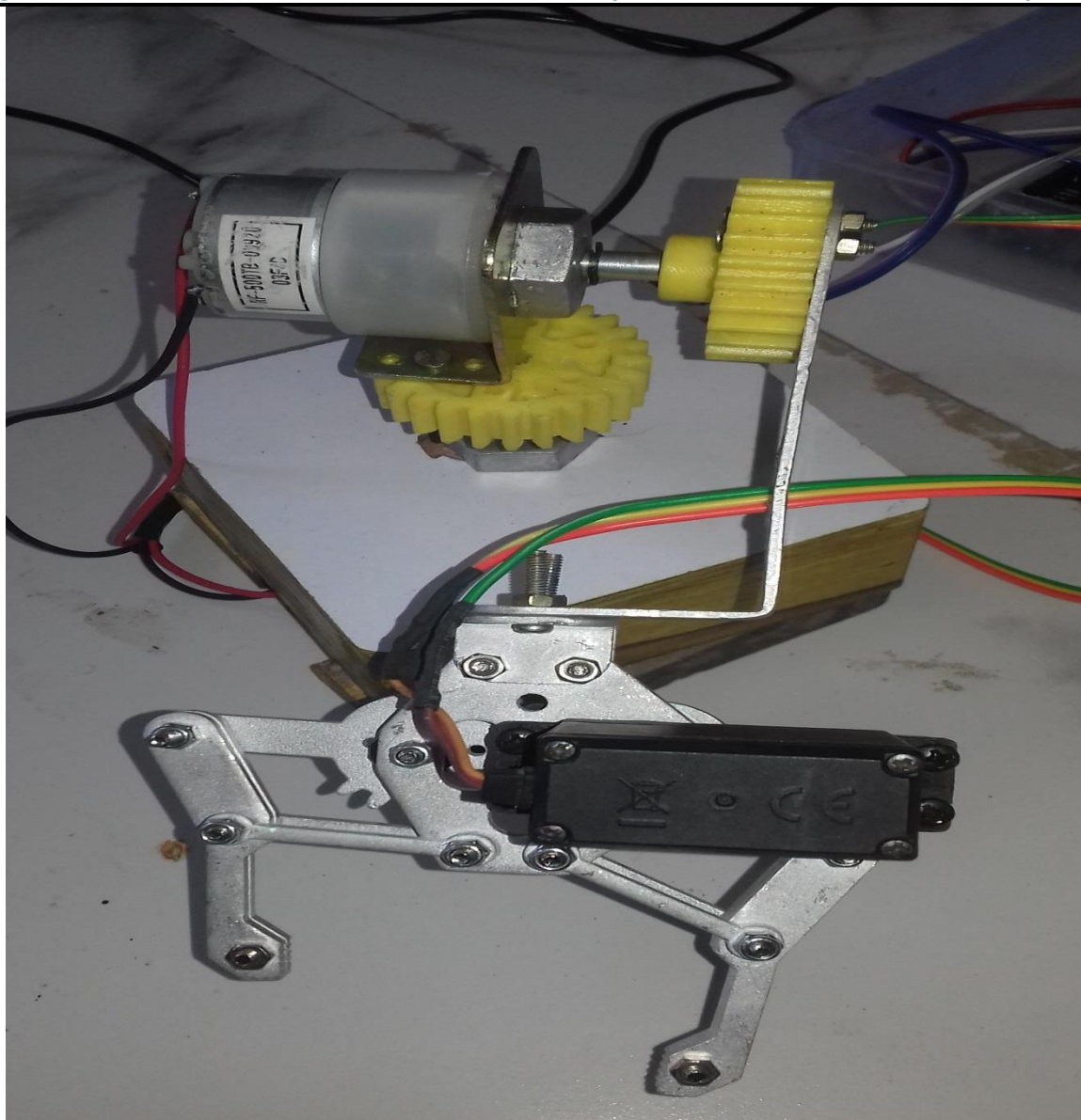
Fig 10 :- Hardware Connections

4) Stepper Motor:-

A **stepper motor** or **step motor** or **stepping motor** is a brushless DC electric motor that divides a full rotation into a number of equal steps. The motor's position can then be commanded to move and hold at one of these steps without any position sensor for feedback (an open-loop controller), as long as the motor is carefully sized to the application in respect to torque and speed.

➤ Fundamental operation:-

Brushed DC motors rotate continuously when DC voltage is applied to their terminals. The stepper motor is known by its property to convert a train of input pulses (typically square wave pulses) into a precisely defined increment in the shaft position. Each pulse moves the shaft through a fixed angle.[3]



Screenshot 1:-Stepper Motor

Stepper motors effectively have multiple "toothed" electromagnets arranged around a central gear-shaped piece of iron. The electromagnets are energized by an external driver circuit or a micro controller. To make the motor shaft turn, first, one electromagnet is given power, which magnetically attracts the gear's teeth. When the gear's teeth are aligned to the first electromagnet, they are slightly offset from the next electromagnet. This means that when the next electromagnet is turned on and the first is turned off, the gear rotates slightly to align with the next one. From there the process is repeated. Each of those rotations is called a "step", with an integer number of steps making a full rotation. In that way, the motor can be turned by a precise angle. The circular arrangement of electromagnets is divided into groups, each group called a phase, and there is an equal number of electromagnets per group. The number of groups is chosen by the designer of the stepper motor. The electromagnets of each group are interleaved with the electromagnets of other groups to form a uniform pattern of arrangement. For example, if the stepper motor has two groups identified as A or B, and ten electromagnets in total, then the grouping pattern would be ABABABABAB. Electromagnets within the same group are all energized together. Because of this, stepper motors with more phases typically have more wires (or leads) to control the motor.[2]

➤ Types:-

There are three main types of stepper motors.

1. Permanent magnet stepper
2. Hybrid synchronous stepper
3. Variable reluctance stepper

Permanent magnet motors use a permanent magnet (PM) in the rotor and operate on the attraction or repulsion between the rotor PM and the stator electromagnets. Variable reluctance (VR) motors have a plain iron rotor and operate based on the principle that minimum reluctance occurs with minimum gap, hence the rotor points are attracted toward the stator magnet poles.

➤ Two phase motor stepper:-

There are two basic winding arrangements for the electromagnetic coils in a two phase stepper motor: bipolar and unipolar.

➤ Unipolar motors:-

A unipolar stepper motor has one winding with center tap per phase. Each section of windings is switched on for each direction of magnetic field. Since in this arrangement a magnetic pole can be reversed without switching the direction of current, the commutation circuit can be made very simple (e.g., a single transistor) for each winding. Typically, given a phase, the center tap of each winding is made common: giving

three leads per phase and six leads for a typical two phase motor. Often, these two phase commons are internally joined, so the motor has only five leads.

A micro controller or stepper motor controller can be used to activate the drive transistors in the right order, and this ease of operation makes unipolar motors popular with hobbyists; they are probably the cheapest way to get precise angular movements.[3]

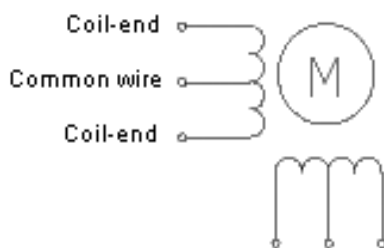


Fig 11 :- Unipolar stepper motor coils

For the experimenter, the windings can be identified by touching the terminal wires together in PM motors. If the terminals of a coil are connected, the shaft becomes harder to turn. One way to distinguish the center tap (common wire) from a coil-end wire is by measuring the resistance. Resistance between common wire and coil-end wire is always half of the resistance between coil-end wires. This is because there is twice the length of coil between the ends and only half from center (common wire) to the end. A quick way to determine if the stepper motor is working is to short circuit every two pairs and try turning the shaft. Whenever a higher than normal resistance is felt, it indicates that the circuit to the particular winding is closed and that the phase is working.[5]

➤ Bipolar Motors:-

Bipolar motors have a single winding per phase. The current in a winding needs to be reversed in order to reverse a magnetic pole, so the driving circuit must be more complicated, typically with an H-bridge arrangement (however there are several off-the-shelf driver chips available to make this a simple affair). There are two leads per phase, none are common. Static friction effects using an H-bridge have been observed with certain drive topologies. Dithering the stepper signal at a higher frequency than the motor can respond to will reduce this "static friction" effect. Because windings are better utilized, they are more powerful than a unipolar motor of the same weight. This is due to the physical space occupied by the windings. A unipolar motor has twice the amount of wire in the same space, but only half used at any point in time, hence is 50% efficient (or approximately 70% of the torque output available). Though a bipolar stepper motor is more complicated to drive, the abundance of driver chips means this is much less difficult to achieve. An 8-lead stepper is wound like a unipolar stepper, but the leads are not joined to common internally to the motor. This kind of motor can be wired in several configurations:[3]

- Unipolar.
- Bipolar with series windings. This gives higher inductance but lower current per winding.
- Bipolar with parallel windings. This requires higher current but can perform better as the winding inductance is reduced.
- Bipolar with a single winding per phase. This method will run the motor on only half the available windings, which will reduce the available low speed torque but require less current

➤ Stepper driver motor circuit:-

Stepper motor performance is strongly dependent on the driver circuit. Torque curves may be extended to greater speeds if the stator poles can be reversed more quickly, the limiting factor being a combination of the winding inductance. To overcome the inductance and switch the windings quickly, one must increase the drive voltage. This leads further to the necessity of limiting the current that these high voltages may otherwise induce.[6]

An additional limitation, often comparable to the effects of inductance, is the back-EMF of the motor. As the motor's rotor turns, a sinusoidal voltage is generated proportional to the speed (step rate). This AC voltage is subtracted from the voltage waveform available to induce a change in the current.[5]

➤ L/R driver circuits:-

L/R driver circuits are also referred to as constant voltage drives because a constant positive or negative voltage is applied to each winding to set the step positions. However, it is winding current, not voltage that applies torque to the stepper motor shaft. The current I in each winding is related to the applied voltage V by the winding inductance L and the winding resistance R . The resistance R determines the maximum current according to Ohm's law $I=V/R$. The inductance L determines the maximum rate of change of the current in the winding according to the formula for an inductor $dI/dt = V/L$. Thus when controlled by an L/R drive, the maximum speed of a stepper motor is limited by its inductance since at some speed, the voltage U will be changing faster than the current I can keep up. In simple terms the rate of change of current is L/R (e.g. a 10 mH inductance with 2 ohms resistance will take 5 ms to reach approx 2/3 of maximum torque or around 24 ms to reach 99% of max torque). To obtain high torque at high speeds requires a large drive voltage with a low resistance and low inductance. With an L/R drive it is possible to control a low voltage resistive motor with a higher voltage drive simply by adding an external resistor in series with each winding. This will waste power in the resistors, and generate heat. It is therefore considered a low performing option, albeit simple and cheap.[3]

Modern voltage-mode drivers overcome some of these limitations by approximating a sinusoidal voltage waveform to the motor phases. The amplitude of the voltage waveform is set up to increase with step rate. If properly tuned, this compensates the effects of inductance and back-EMF, allowing decent performance relative to current-mode drivers, but at the expense of design effort (tuning procedures) that are simpler for current-mode drivers.[8]

➤ Chopper drive circuits:-

Chopper drive circuits are referred to as constant current drives because they generate a somewhat constant current in each winding rather than applying a constant voltage. On each new step, a very high voltage is applied to the winding initially. This causes the current in the winding to rise quickly since $di/dt = V/L$ where V is very large. The current in each winding is monitored by the controller, usually by measuring the voltage across a small sense resistor in series with each winding. When the current exceeds a specified current limit, the voltage is turned off or "chopped", typically using power transistors. When the winding current drops below the specified limit, the voltage is turned on again. In this way, the current is held relatively constant for a particular step position. This requires additional electronics to sense winding currents, and control the switching, but it allows stepper motors to be driven with higher torque at higher speeds than L/R drives. Integrated electronics for this purpose are widely available.[8]

➤ Stepper motor rating and specifications:-

Stepper motors' nameplates typically give only the winding current and occasionally the voltage and winding resistance. The rated voltage will produce the rated winding current at DC: but this is mostly a meaningless rating, as all modern drivers are current limiting and the drive voltages greatly exceed the motor rated voltage.[1]

Data sheets from the manufacturer often indicate Inductance. Back-EMF is equally relevant, but seldom listed (it is straightforward to measure with an oscilloscope). These figures can be helpful for more in-depth electronics design, when deviating from standard supply voltages, adapting third party driver electronics, or gaining insight when choosing between motor models with otherwise similar size, voltage, and torque specifications.

A stepper's low speed torque will vary directly with current. How quickly the torque falls off at faster speeds depends on the winding inductance and the drive circuitry it is attached to, especially the driving voltage.[2]

Steppers should be sized according to published torque curve, which is specified by the manufacturer at particular drive voltages or using their own drive circuitry. Dips in the torque curve suggest possible resonances, whose impact on the application should be understood by designers.[6]

Step motors adapted to harsh environments are often referred to as IP65 rated

The US National Electrical Manufacturers Association (NEMA) standardises various aspects of stepper motors. They are typically referred with NEMA DD, where DD is the diameter of the faceplate in inches $\times 10$ (e.g., NEMA 17 has diameter of 1.7 inches). There are further specifiers to describe stepper motors, and such details may be found in the ICS 16-2001 standard (section 4.3.1.1). [5]

➤ Applications:-

Computer controlled stepper motors are a type of motion-control positioning system. They are typically digitally controlled as part of an open loop system for use in holding or positioning applications.[7]

In the field of lasers and optics they are frequently used in precision positioning equipment such as linear actuators, linear stages, rotation stages, goniometers, and mirror mounts. Other uses are in packaging machinery, and positioning of valve pilot stages for fluid control systems.[7]

Commercially, stepper motors are used in floppy disk drives, flatbed scanners, computer printers, plotters, slot machines, image scanners, compact disc drives, intelligent lighting, camera lenses, CNC machines and, more recently, in 3D printers.[6]

➤ stepper motor system:-

stepper motor system consists of three basic elements, often combined with some type of user interface (host computer, PLC or dumb terminal):

✓ Indexers

The indexer (or controller) is a microprocessor capable of generating step pulses and direction signals for the driver. In addition, the indexer is typically required to perform many other sophisticated command functions.

✓ Drivers

The driver (or amplifier) converts the indexer command signals into the power necessary to energize the motor windings. There are numerous types of drivers, with different voltage and current ratings and construction technology. Not all drivers are suitable to run all motors, so when designing a motion control system the driver selection process is critical.

✓ Stepper motors

The stepper motor is an electromagnetic device that converts digital pulses into mechanical shaft rotation. Advantages of step motors are low cost, high reliability, high torque at low speeds and a simple, rugged construction that operates in almost any environment. The main disadvantages in using a stepper motor is the resonance effect often exhibited at low speeds and decreasing torque with increasing speed.[5]

➤ Advantages:-

- Low cost for control achieved
- High torque at startup and low speeds
- Ruggedness
- Simplicity of construction
- Can operate in an open loop control system
- Low maintenance
- Less likely to stall or slip
- Will work in any environment
- Can be used in robotics in a wide scale.
- High reliability
- The rotation angle of the motor is proportional to the input pulse.
- The motor has full torque at standstill (if the windings are energized)
- Precise positioning and repeatability of movement since good stepper motors have an accuracy of 3–5% of a step and this error is non-cumulative from one step to the next.
- Excellent response to starting/stopping/reversing.

- Very reliable since there are no contact brushes in the motor. Therefore, the life of the motor is simply dependent on the life of the bearing.
- The motors response to digital input pulses provides open-loop control, making the motor simpler and less costly to control.
- It is possible to achieve very low-speed synchronous rotation with a load that is directly coupled to the shaft.
- A wide range of rotational speeds can be realized as the speed is proportional to the frequency of the input pulses.

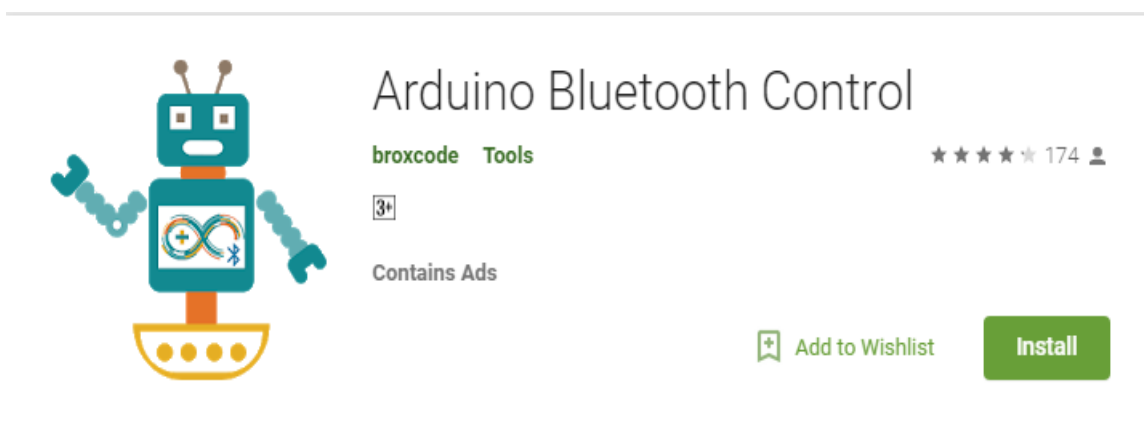
5) Android mobile:-

Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.[3]

Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the first commercial Android device launched in September 2008. The operating system has since gone through multiple major releases, with the current version being 8.1 "Oreo", released in December 2017. The core Android source code is known as **Android Open Source Project (AOSP)**, and is primarily licensed under the Apache License.[5]

Android has been the best-selling OS worldwide on smartphones since 2011 and on tablets since 2013. As of May 2017, it has over two billion monthly active users, the largest installed base of any operating system, and as of 2017, the Google Play store features over 3.5 million apps.[6]

Android app (Bluetooth controller app):-



Screenshot 2 :-Bluetooth control app

Arduino Bluetooth Control is an application that allows you to control your arduino board (and similar boards) via Bluetooth, and so to create awesome and fully customized projects, with the new features available within the app. The settings section allows you to adapt the application to your needs, through a very simple and intuitive interface.

Once launched and connected to your module, the application handles sms receiving to control your board from wherever you are. Everything you need to do is just to keep a connected phone near your board via the application, and send an sms to it that starts with the word Arduino and followed by the data you need to send. For example -Arduino 1- means that -1- will be sent to your board, - Arduino hello- would send -hello- and so on.

The application also smartly remembers your bluetooth module and will always try to connect automatically to the latest one you have used, so you won't have to select it every time you use it .[8]

1. Metrics tool :-

This tool was optimized to receive data via the `println()` function of arduino, which allows special processing of the data received, like in the "Metrics" tool. It allows you to receive only numbers and fix alarms to get notified about the variations of the value received, either by classic notifications or by sms to another phone, you will only need to specify the number in the "Settings" section. You can also do the same the other way around, as mentioned before. Once the alarm triggered, a stop button shows up, allowing you to stop it. Besides you can activate the shaking mode, that will allow you to send data simply by shaking your phone.

2. Arrow keys :-

This tool provides direction buttons that can fully customized with the data to send, and the sensitivity, which allows to send continuously data to the board by maintaining long press on them.

3. Terminal:-

This tool is just a classic terminal that receives and sends data to the board, displayed with the timestamp corresponding to each action.

4. Buttons and slider :-

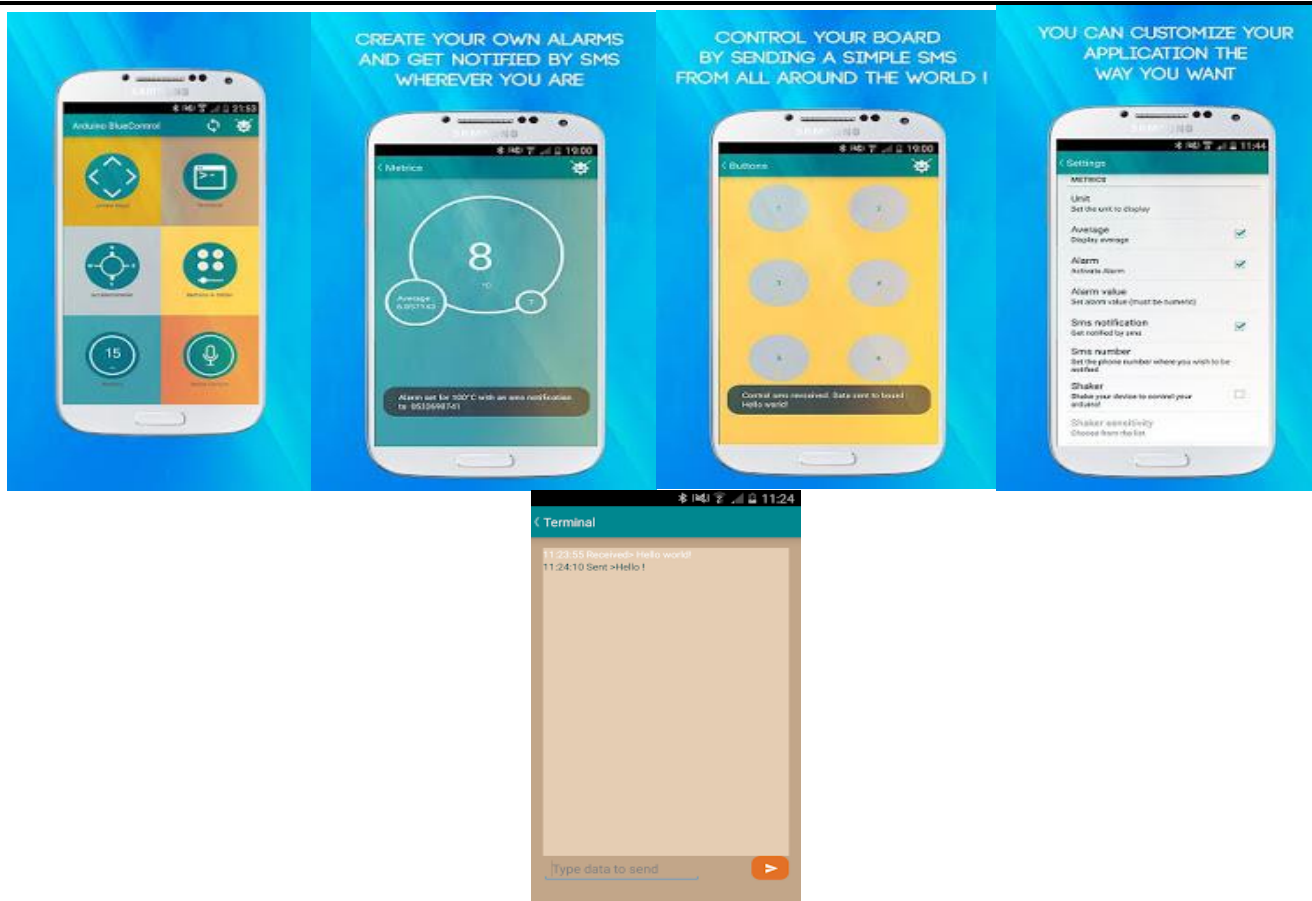
In portrait orientation, this tool provides 6 buttons fully customized, that will allow you to send specific data when pressed. When you rotate your device, a slider view shows up, to which you can set the range of the data to be sent.

5. Accelerometer :-

This tool permits you to interpret the gesture commands of your phone, and send the corresponding data to your board, and so, your phone can be the steering wheel of your robot. You can of course set the sensitivity of it through the settings interface.

6. Voice Control :-

Have you ever dreamed of talking to you robots? Well now your dream is becoming true! With Arduino Bluetooth Control, you can customize your own vocal commands and use them to control all your microcontroller-based boards



Screenshot 3 :-Working of Bluetooth controller app

CHAPTER 3 Software Design And Description

This chapter describes the software that is being used in the project.

3.1 Software Requirements:-

1.Arduino:-

The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself (DIY) kits.[5]



Arduino Tutorials

CoderBro Education

★★★★★ 362

3+

Contains Ads

Add to Wishlist

Install

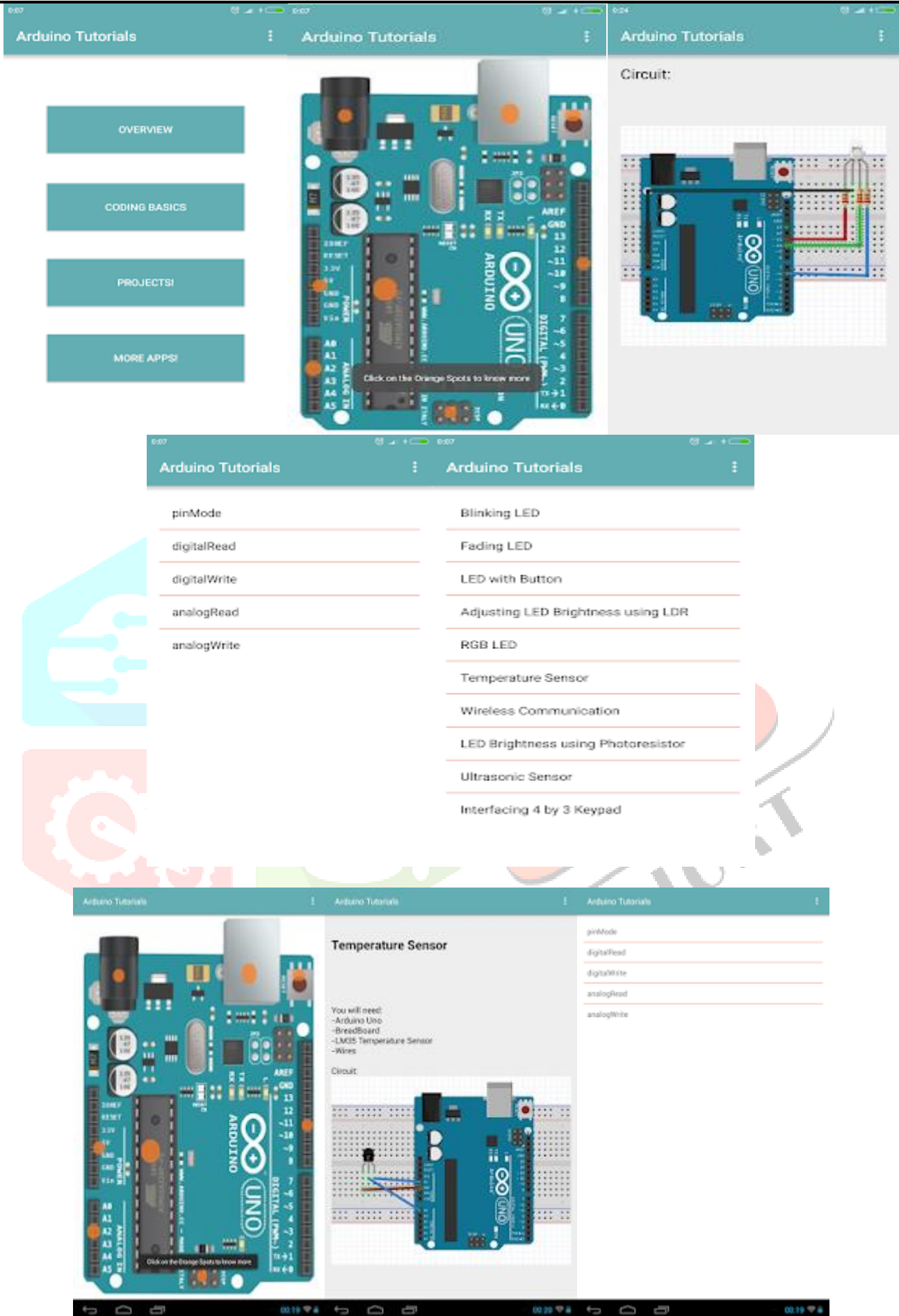
Screenshot 4 :-Arduino app

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or Breadboards (shields) and other circuits. [6]

The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.[7]

The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.[8]

The name Arduino comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.[1]



Screenshot 5:- Arduino App

✚ Coding for software used in project:-

```

chikhali_project | Arduino 1.5.8
File Edit Sketch Tools Help

chikhali_project
String voice;
#include <Servo.h>
Servo myservo;
int pos = 0;

const int m1=A0;
const int m2=A1;

const int m3=A2;
const int m4=A3;

void setup()
{
  Serial.begin(9600);
  myservo.attach(9);

  pinMode(m1,OUTPUT);
  pinMode(m2,OUTPUT);
  pinMode(m3,OUTPUT);
  pinMode(m4,OUTPUT);

  digitalWrite(m1,LOW);
  digitalWrite(m2,LOW);
  digitalWrite(m3,LOW);
  digitalWrite(m4,LOW);
}

```

```

chikhali_project
}

void loop()
{
  while (Serial.available())
  {
    delay(10);
    char c = Serial.read();

    if (c == '#') {break;}
    }
    voice += c;
  }
  if (voice.length() > 0)
  {
    Serial.println(voice);

    if(voice == "1")
    {
      digitalWrite(m1,LOW);
      digitalWrite(m2,HIGH);
      delay(100);
      digitalWrite(m1,LOW);
      digitalWrite(m2,LOW);
    }
  }
}

```

```

chikhali_project | Arduino 1.5.8
File Edit Sketch Tools Help

chikhali_project
    if(voice == "2")
    {
      digitalWrite(m1,HIGH);
      digitalWrite(m2,LOW);
      delay(200);
      digitalWrite(m1,LOW);
      digitalWrite(m2,LOW);
    }

    if(voice == "4")
    {
      digitalWrite(m3,LOW);
      digitalWrite(m4,HIGH);
      delay(100);
      digitalWrite(m3,LOW);
      digitalWrite(m4,LOW);
    }

    if(voice == "3")
    {
      digitalWrite(m3,HIGH);
      digitalWrite(m4,LOW);
      delay(100);
      digitalWrite(m3,LOW);
      digitalWrite(m4,LOW);
    }
}

```

```

if(voice == "5")
{
myservo.write(230);           // tell servo to go to position in variable 'pos'
delay(15);
}

if(voice == "6")
{
myservo.write(0);           // tell servo to go to position in variable 'pos'
delay(15);
}

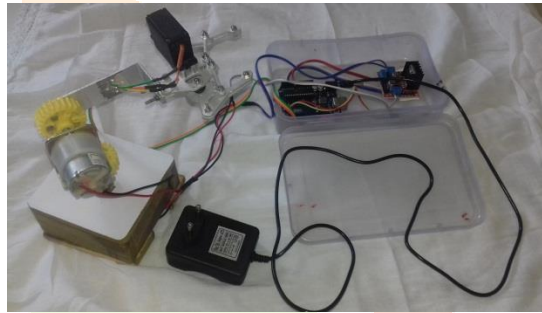
voice="";
}
}
    
```

Screenshots 6 :-Programming for software

**CHAPTER 4
Implementation**

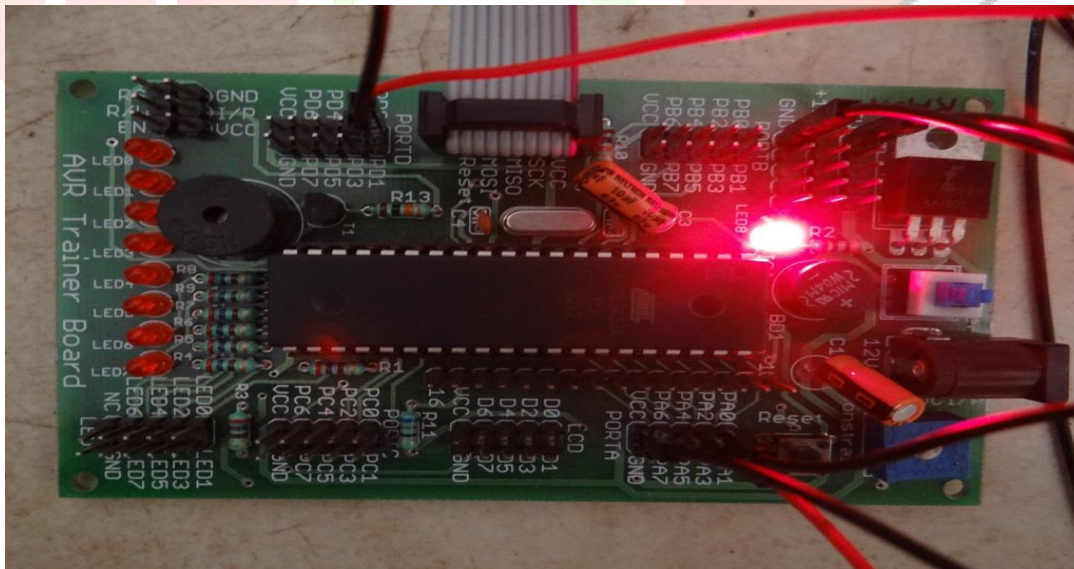
4.1 Implementation:-

➤ Connection of hardware:-

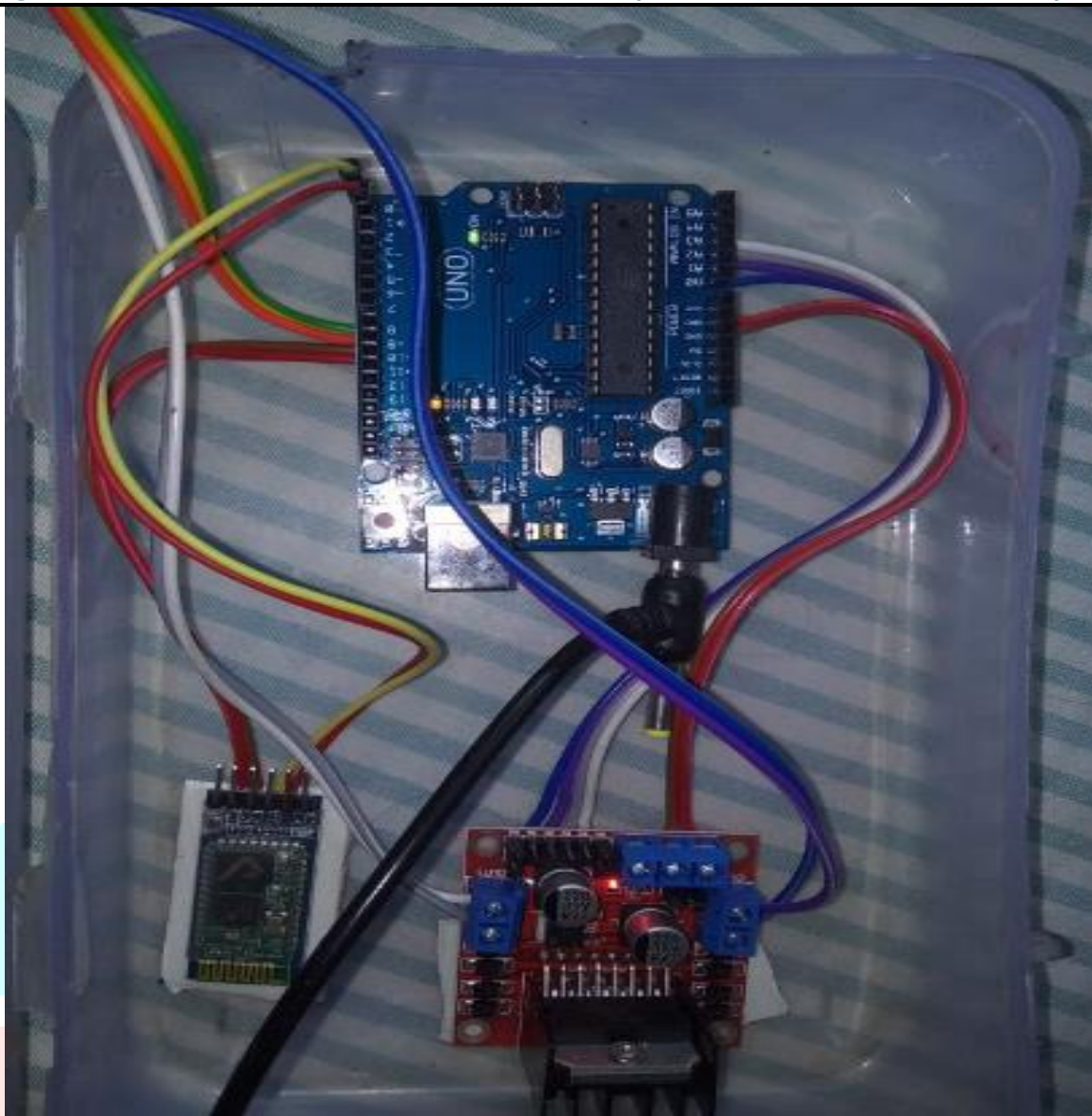


Screenshot 7 :- Connection of hardware

➤ The ATmega328 development boards are shown in Figure:-



➤ Connection of ATmega328 arduino uno,L293D Ic Motor Driver, Bluetooth HC-05,stepper motor 3



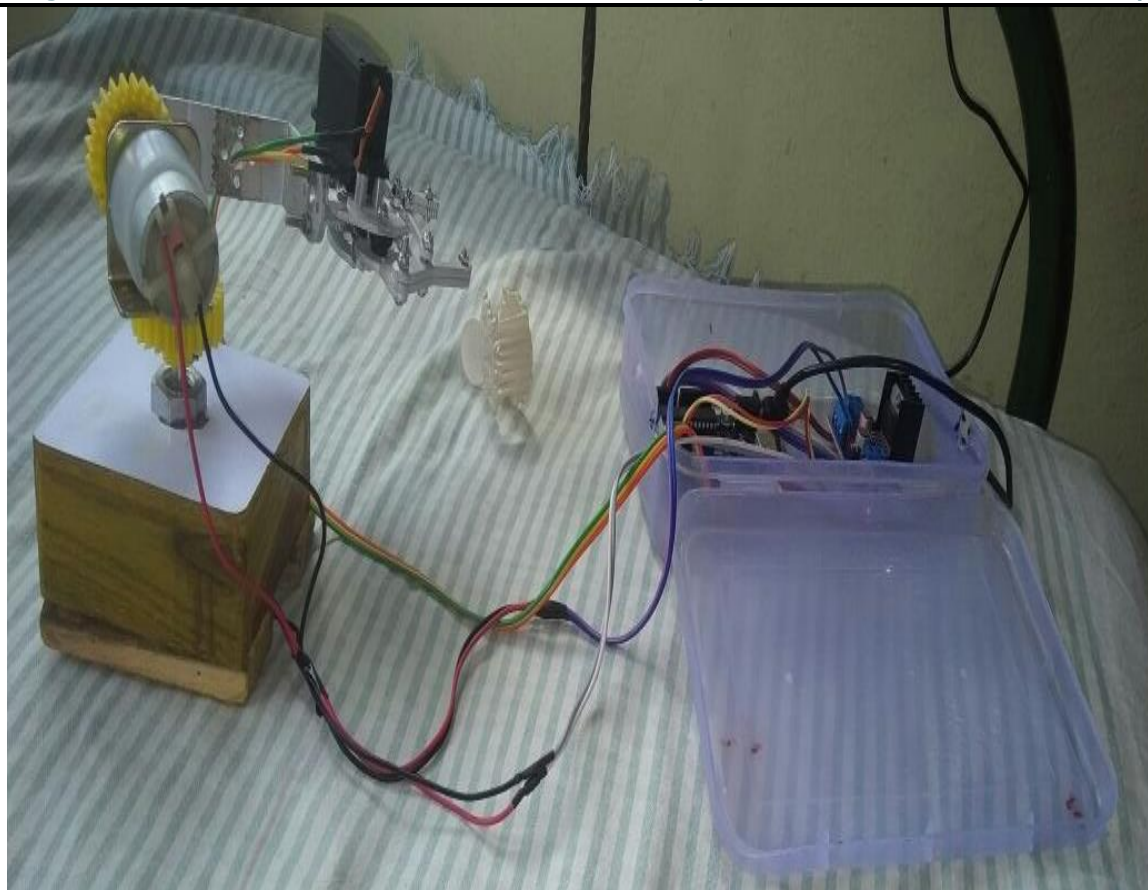
Screenshot 8 :- Connection of ATmega328 arduino uno,L293D Ic Motor Driver, Bluetooth HC-05,stepper motor

Working Procedure :-

In this project, we are using three geared DC motors each of which can be controlled by L293D motor driver. Here, two reference positions are chosen. First reference position is the place from where the arm has to pick the object and second reference position is the place where the robot has to place the object. First the microcontroller signals the motor-3 via driver circuit one to make the rotation of the arm to the desired direction. Then the signal from microcontroller is given to driver circuit1 to drive the 2nd motor so that it can make up and down movement. Next motor 1 which is situated at the gripper is activated so that gripper holds the object. Next, motor-3 is again activated to turn the motor towards destination direction, motor is then activated to make the down movement of the arm and finally, gripper motor is activated to release the object.[3]



Screenshots 9 :-Pick the object



Screenshots 10 : Drop the object

CHAPTER 5 Future Scope

- ✓ Robotic Arms has a wide scope of development. In the near future the arms will be able to perform every task as humans and in much better way. Imagination is the limit for its future applications.
- ✓ It can be a real boon for handicapped people, who are paralyzed or lost their hands in some accident. The arm can be trained to listen to the command from a human and perform that task. A Precise gesture controlled system is also possible. Wearable devices can be used to send the command and control the movements of the arm.
- ✓ Brain Computer Interface (BCI) is an immerging field of research. BCI can be used to acquire signals from the human brain and control the arm. The system can work in the same way as human arm. A person who may have lost his hand in any accident can resume his life like previous by such artificial arms. Robotic arms are versatile and have enormous ways of implementations.

CHAPTER 6 Conclusion

- ✓ This paper presents our study on early research and developments in robotic arms. The objective of project has been achieved which was developing the hardware and software for an controlled robotic arm.
- ✓ The main aim of our project is to build a robotic arm that can be pick things.
- ✓ Even after many years of research, the applications of robotic arm are restricted to the industries and primarily used in manufacturing units for increasing productivity. These arms are very sophisticated and can manage to make extremely precise movements. The robotic arms have wide variety of general purpose and domestic applications too, which are not much explored. The robotic arms will be available everywhere to serve as a “helping hand”.
- ✓ From observation that has been made, it clearly shows that its movement is precise, accurate and easy to control and user friendly to use.
- ✓ The robotic arms develop successfully as the movement of the robotic arm can be controlled precisely.
- ✓ The robotic arm controlled method is accepted to overcome the problem such as placing or picking object that away from the user, pick and place object in very fast and easy manner.

References

- [1] Moshe Shoham: Textbook of Robotics: I Basic Concepts. Anchor Brendon Ltd.
- [2] Richard D. Klafter: Robotic Engineering. Prentice-Hall of India.
- [3] The development of six D.O.F. robot arm for intelligent robot Jie-Tong Zou ; Dept. of Aeronaut. Eng., Nat. Formosa Univ., Huwei Township, Taiwan; Des-Hun Tu.
- [4] Atmel ATmega328 Datasheet, AVR Corporation, Feb 2014

- [5] Mohd Ashiq Kamaril Yusoffa, Reza Ezuan Saminb, Babul Salam Kader Ibrahimc, “ Robotic Arm”, International Symposium on Robotics 2012 (IRIS 2015), July 2015
- [6] Wan Muhamad Hanif Wan Kadir, Reza Ezuan Samin, Babul Salam Kader Ibrahim, “Bluetooth Controller Robotic Arm”. International Symposium on Robotics 2012(IRIS 2012), July 2012
- [7] The Multiple-Function Intelligent Robotic Arms Rong-Jyue Wang, Jun-Wei Zhang, Jia-Ming Xu, and Hsin-Yu Liu
- [8] Avinash, “Microcontrollers robotic arm” , <http://extremeelectronics.co.in/avrtutorials/using-the-usart-of-avr-microcontrollers/>, December 2013

