**CHAPTER 1**

**INTRODUCTION TO PROJECT**

* 1. **INTRODUCTION**

Robot is an electromechanical machine and used for various purposes in industrial and domestic applications. Robot appliances are entering in the consumer market, since the introduction of iRobots. Many related appliances from various companies have been followed. Initially the main focus was on having a cleaning device. As the time pass on many improvements were made and more efficient appliances were developed. In early, 2010 a new automatic floor cleaner robot “Mint” was developed by Jen Steffen. Detachable clothes were attached for sweeping and mopping purposes.

In this research work a floor cleaner robot based on Atmega32 have been developed. This cleaner robot is an electric home appliance . Unlike other floor cleaner robots this is not a vacuum cleaner robot; it performs sweeping and mopping operation. Detachable mop is used for mopping. It works on 12V supply. In the automatic mode, robot performs all operations itself.

Firstly robot starts it moves forward and perform cleaning action. To make whole system wireless, Bluetooth modules have been used in automatic and manual with 50m range. For user convenience automatic water sprayer is attached which automatically spray water for mopping, therefore no need to attach wet cloth again and again for mopping . Fan is used to dry the wet floor.

Motor driver circuit have been used to drive the motors. Four motors have been used to perform respected operations like to move the robot, for water pump, for cleaner. Relays have been used to drive the water pump and cleaner motor.

LM293D IC has been used to drive wheel motor. In the manual mode, user itself operates the robot. Bluetooth module have been used to transmit and receive the signal to operate the robot through remote.

Movement of robot is controlled by user itself through the android app . Therefore user can move the robot in the desired direction. All the information displayed on the LCD.

* 1. **SALIENT FEATURES**

This floor cleaning robot has following salient features which make it a useful tool in today’s world

* **AFFORDABILITY** – Unless like many cleaning devices like vaccum cleaner which will cost upto 10,000 -20,000 and will be heavy for your pocket . The cost of floor cleaning robot will not exceed more than 2,000-2,500 providing you an economical solution for better cleaning .
* **MODIFICATION POSSIBLE** – This project is easily modified and can be used for future advancement by implementing alarm system and obstacle avoiding . Modification can be easily done on the existing model.
* **CLEANING WITH BETTER PROFICIENCY –** Cleaning done by the robots will be much more cleaner and accurate than the cleaning perform by the human hands .
  1. **ADVANTAGES**
* It reduces human energy and efforts.
* People in cities have irregular and long working times. In such a situation a person will always find ways of saving time.
* Helping physically disable person is also advantage of this project .Automatic mode of this robot helps physical disable person.
* We can use this robot in Automatic and Manual mode also
* Easy mounting and easy to operate .Due to that, it is user friendly.
  1. **APPLICATIONS**
* Main purpose of this project is Cleaning.
* We can save our time by using this robot.
* Able to go under furniture and around corners

**CHAPTER 2**

**HARDWARE DESCRIPTION**

**2.1DESCRIPTION**

This floor cleaning robot is consist of atmega32 which will contain all the coding and instructions that are required to make our robot work . Bluetooth module will help us to connect our smartphone with this robot and control its motion . L298D or Motor Driver IC will use to control the motion or speed of robot .

**2.2 COMPONENTS USED**

Many semiconductors and electronics devices are connected together for the proper working of this project . The list of components used in this project is as follows :

1. Atmega 32
2. Bluetooth Module
3. Motor driver IC
4. Water Pump
5. DC Gear Motor
6. LED
7. Resistor
8. Voltage Regulator IC
9. 12V DC Battery
10. Capacitor
11. SPST Switch

**2.3 ATMEGA 32**

ATMEGA328P is high performance, low power controller from Microchip. ATMEGA328P is an 8-bit microcontroller based on AVR RISC architecture. It is the most popular of all AVR controllers as it is used in ARDUINO boards.

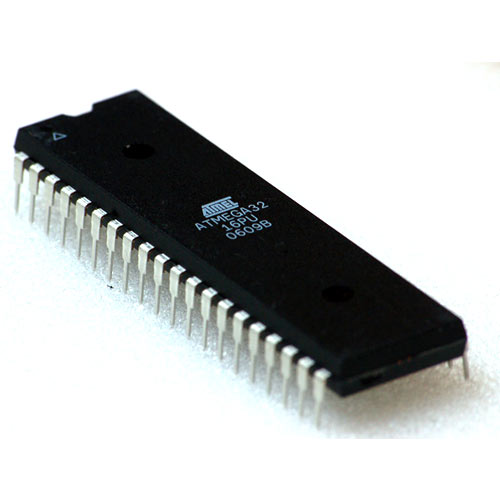


Fig. 2.1 : ATMega 328P Microcontroller

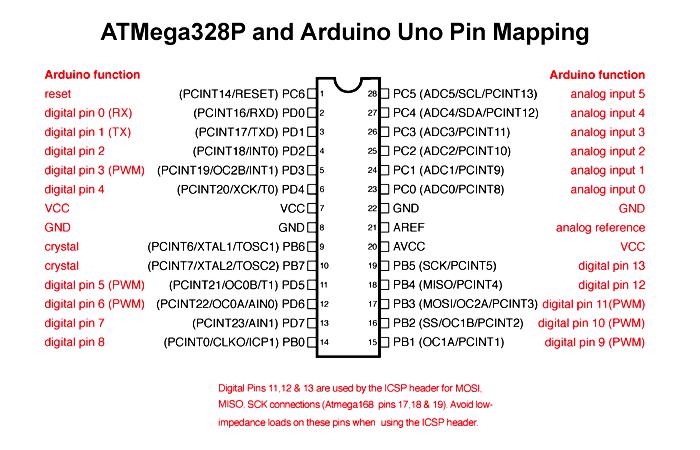
****

Fig. 2.2 : ATMega 32 Pin description

**2.3.1 ATMEGA 32 PIN DESCRIPTION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pin No.** | **Pin name** | **Description** | | **Alternate Function** |
| 1 | PB0(XCK/T0) | Pin 0 of  PORTB | | T0( Timer0 External Counter Input)  XCK ( USART External Clock I/O) |
| 2 | PB1(T1) | Pin 1 of  PORTB | | T1(Timer1 External Counter Input) |
| 3 | PB2(INT2/AIN0) | Pin 2 of  PORTB | | AIN0(Analog Comparator Positive I/P)  INT2( External Interrupt 2 Input) |
| 4 | PB3(OC0/AIN1) | Pin 3 of  PORTB | | AIN1(Analog Comparator Negative I/P)  OC0 (Timer0 Output Compare Match Output) |
| 5 | PB4(SS) | Pin 4 of  PORTB | | SS (SPI Slave Select Input).  This pin is low when controller acts as slave.  [Serial Peripheral Interface (SPI) for programming] |
| 6 | PB5(MOSI) | Pin 5 of  PORTB | | MOSI (Master Output Slave Input). When controller acts      as slave, the data is received by this pin.  [Serial Peripheral Interface (SPI) for programming] |
| 7 | PB6(MISO) | Pin 6 of  PORTB | | MISO (Master Input Slave Output). When controller acts as slave, the data is sent to master by this controller through this pin.  [Serial Peripheral Interface (SPI) for programming] |
| 8 | PB7(SCK) | Pin 7 of  PORTB | | SCK (SPI Bus Serial Clock). This is the clock shared between this controller and other system for accurate data transfer.  [Serial Peripheral Interface (SPI) for programming] |
| 9 | RESET | Reset Pin, Active Low Reset | | Pulled HIGH to RESET controller. |
| 10 | Vcc | Vcc = +5V | |  |
| 11 | GND | GROUND | |  |
| 12 | XTAL2 | Connected to Crystal Oscillator | |  |
| 13 | XTAL1 | Connected to Crystal Oscillator | |  |
| 14 | PD0(RXD) | Pin 0 of  PORTD | | RXD (USART Input Pin)  USART Serial Communication Interface  [Can be used for programming] |
| 15 | PD1(TXD) | Pin 1 of  PORTD | | TXD (USART Output Pin)  USART Serial Communication Interface  [Can be used for programming] |
| 16 | PD2(INT0) | Pin 2 of  PORTD | | External Interrupt INT0 |
| 17 | PD3(INT1) | Pin 3 of  PORTD | | External Interrupt INT1 |
| 18 | PD4(OC1B) | Pin 4 of  PORTD | | PWM Channel Outputs |
| 19 | PD5(OC1A) | Pin 5 of  PORTD | |
| 20 | PD6(ICP) | Pin 6 of  PORTD | | Timer/Counter1 Input Capture Pin |
| 21 | PD7 (OC2) | Pin 7 of  PORTD | | Timer/Counter2 Output Compare Match Output |
| 22 | PC0 (SCL) | Pin 0 of  PORTC | | TWI Interface |
| 23 | PC1 (SDA) | Pin 1 of  PORTC | |
| 24 | PC2 (TCK) | Pin 2 of  PORTC | | JTAG Interface |
| 25 | PC3 (TMS) | Pin 3 of  PORTC | |
| 26 | PC4 (TDO) | Pin 4 of  PORTC | |
| 27 | PC5 (TDI) | Pin 5 of  PORTC | |
| 28 | PC6 (TOSC1) | Pin 6 of  PORTC | | Timer Oscillator Pin 1 |
| 29 | PC7 (TOSC2) | Pin 7 of  PORTC | | Timer Oscillator Pin 2 |
| 30 | AVcc | Vcc for Internal ADC  Converter | | |
| 31 | GND | GROUND | | |
| 32 | AREF | Analog Reference Pin for ADC | | |
| 33 | PA7 (ADC7) | Pin 7 of  PORTA | ADC (Analog to Digital Converter) Channel 7 | |
| 34 | PA6 (ADC6) | Pin 6 of  PORTA | ADC (Analog to Digital Converter) Channel 6 | |
| 35 | PA5 (ADC5) | Pin 5 of  PORTA | ADC (Analog to Digital Converter) Channel 5 | |
| 36 | PA4 (ADC4) | Pin 4 of  PORTA | ADC (Analog to Digital Converter) Channel 4 | |
| 37 | PA3 (ADC3) | Pin 3 of  PORTA | ADC (Analog to Digital Converter) Channel 3 | |
| 38 | PA2 (ADC2) | Pin 2 of  PORTA | ADC (Analog to Digital Converter) Channel 2 | |
| 39 | PA1 (ADC1) | Pin 1 of  PORTA | ADC (Analog to Digital Converter) Channel 1 | |
| 40 | PA0 (ADC0) | Pin 0 of  PORTA | ADC (Analog to Digital Converter) Channel 0 | |
|  |  |  |  |  |

**2.3.2 ATMEGA32 FEATURES**

|  |  |
| --- | --- |
| **ATMEGA32 – Simplified Features** | |
| CPU | 8-bit AVR |
| Number of Pins | 40 |
| Operating Voltage (V) | +4.5 to +5.5 V (+5.5V being absolute maximum) |
| Number of I/O pins | 32 |
| Communication Interface | JTAG Interface(24,25,26,27 PINS)[Can be used for programming this controller]  Master/Slave SPI Serial Interface(5,6,7,8 PINS) [Can be used for programming this controller]  Programmable Serial USART(14,15 PINS) [Can be used for programming this controller]  Two-wire Serial Interface(22,23 PINS)[Can be used to connect peripheral devices like sensors and LCDs] |
| ADC Module | 8 channels , 10-bit resolution ADC |
| Timer Module | Two 8-bit counters, One 16-bit counter [Total three] |
| Analog Comparators | 1 |
| DAC Module | Nil |
| PWM channels | 4 |
| External Oscillator | 0-8MHz for ATMEGA32L  0-16MHz for ATMEGA32 |
| Internal Oscillator | 0-8MHz  Calibrated Internal Oscillator |
| Program Memory Type | Flash |
| Program Memory (KB) | 32Kbytes[10000 write/erase cycles] |
| CPU Speed (MIPS) | 16 MIPS |
| RAM Bytes | 2Kbytes |
| Data EEPROM | 1024 Bytes |
| Watchdog Timer | Programmable Watchdog Timer with Separate On-chip  Oscillator |
| Power Save Modes | Six Modes[Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby] |
| Operating Temperature | -55°C to +125°C(+125 being absolute maximum, -55 being absolute minimum) |

**2.3.3 APPLICATIONS**

There are thousands of applications for ATMEGA32.

* Temperature control systems
* Analog signal measuring and manipulations.
* Embedded systems like coffee machine, vending machine.
* Motor control systems.
* Digital signal processing.
* Peripheral Interface system.

**2.4 BLUETOOTH MODULE**

* It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard and many more consumer applications.
* It has range up to <100m which depends upon transmitter and receiver, atmosphere, geographic & urban conditions.
* It is IEEE 802.15.1 standardized protocol, through which one can build wireless Personal Area Network ([PAN](https://en.wikipedia.org/wiki/Personal_area_network)). It uses frequency-hopping spread spectrum ([FHSS](https://en.wikipedia.org/wiki/Frequency-hopping_spread_spectrum)) radio technology to send data over air.



Fig. 2.3 : HC-05 Bluetooth Module

* It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART)
* HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.

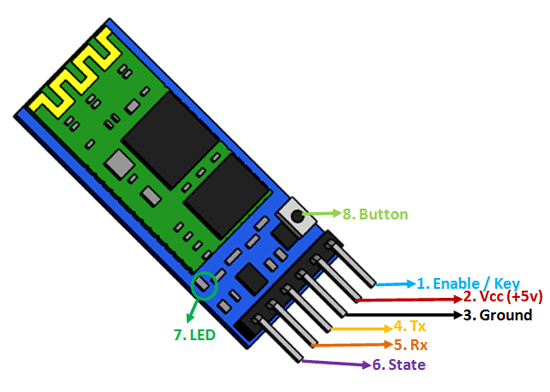


Fig. 2.4 : HC-05 Bluetooth Module Pinout

**2.4.1 PIN CONFIGURATION OF BLUETOOTH MODULE**

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Enable / Key | This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode |
| 2 | Vcc | Powers the module. Connect to +5V Supply voltage |
| 3 | Ground | Ground pin of module, connect to system ground. |
| 4 | TX – Transmitter | Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data. |
| 5 | RX – Receiver | Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth |
| 6 | State | The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly. |
| 7 | LED | Indicates the status of Module   * Blink once in 2 sec: Module has entered Command Mode * Repeated Blinking: Waiting for connection in Data Mode * Blink twice in 1 sec: Connection successful in Data Mode |
| 8 | Button | Used to control the Key/Enable pin to toggle between Data and command Mode |

### ****2.4.2 APPLICATIONS****

* + Wireless communication between two microcontrollers
  + Communicate with Laptop, Desktops and mobile phones
  + Data Logging application
  + Consumer applications
  + Wireless Robots
  + Home Automation

**2.5 MOTOR DRIVER IC L293D**

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](https://www.rakeshmondal.info/High-Torque-Motor-Low-RPM-Motor) with a single L293D IC.



Fig. 2.5 : L293D IC

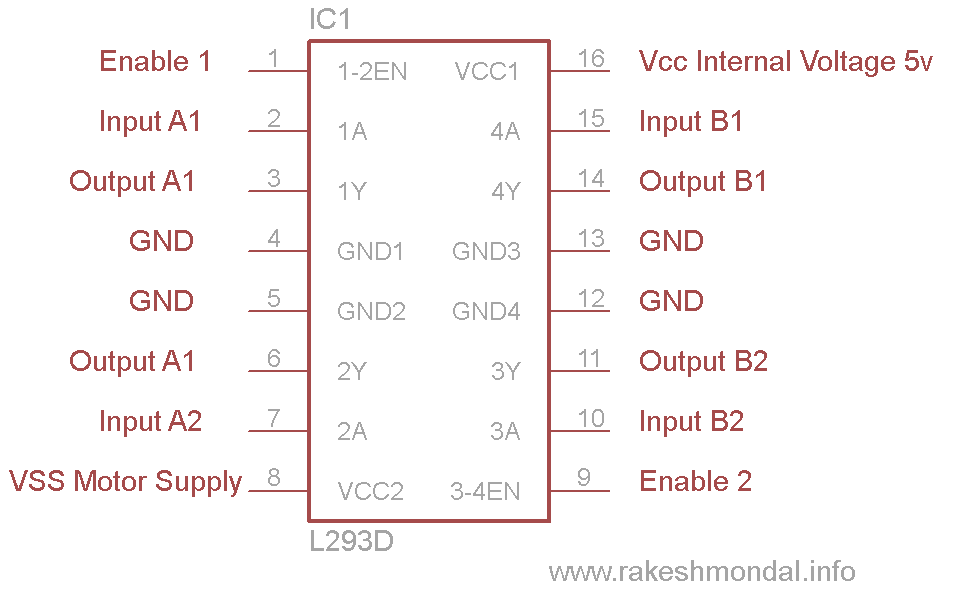


Fig. 2.6 : L293D Pin Configuration

## 2.5.1 WORKING OF L293D

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.

In simple you need to provide Logic 0 or 1 across the input pins for rotating the motor.

## 2.5.2 L293D LOGIC TABLE.

Lets consider a Motor connected on left side output pins (pin 3,6). For rotating the motor in clockwise direction the input pins has to be provided with Logic 1 and Logic 0.

• **Pin 2** = **Logic 1** and **Pin 7**= **Logic 0** | Clockwise Direction  
• **Pin 2** = **Logic 0**and **Pin 7**= **Logic 1** | Anticlockwise Direction  
•**Pin 2**= **Logic 0** and **Pin 7** = **Logic 0** | Idle [No rotation] [Hi-Impedance state]  
• **Pin 2**= **Logic 1** and **Pin 7** = **Logic 1** | Idle [No rotation]

In a very similar way the motor can also operate across input pin 15,10 for motor on the right hand side.

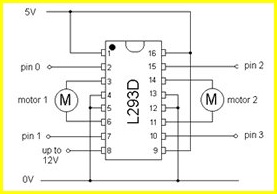


Fig. 2.7 : Circuit Diagram for L293D Motor Driver IC Controller

## 2.5.3 VOLTAGE SPECIFICATION

VCC is the voltage that it needs for its own internal operation 5v; L293D will not use this voltage for driving the motor. For driving the motors it has a separate provision to provide motor supply VSS (V supply).  L293d will use this to drive the motor. It means if you want to operate a motor at 9V then you need to provide a Supply of 9V across VSS Motor supply.

The maximum voltage for VSS motor supply is 36V. It can supply a max current of 600mA per channel. Since it can drive motors Up to 36v hence you can drive pretty big motors with this l293d.VCC pin 16 is the voltage for its own internal Operation. The maximum voltage ranges from 5v and up to 36v.

**2.6 WATER PUMP**

A pump is a device that moves fluids (liquids or gases), or sometimes slurries, by mechanical action. This motor is small, compact and light. It can be controlled from a micro controller /Arduino using our DC Motor Drivers or one of our Relay Boards. You may use our 5V SMPS Power Supply Adapter to run this pump. You may also use our 6V Solar Panel to run the pump with appropriate a 6V voltage regulator.



Fig. 2.8 : 5V Water Pump

**2.6.1 FEATURES**

* Operating DC Voltage: 2.5-6V
* Maximum Water lift height: 40-110cm / 15.75"-43.4"
* Flow rate: 80-120L/H
* Outer Diameter of Water Outlet: 7.5mm / 0.3"
* Inside Diameter of Water Outlet: 5mm / 0.2"
* Pump Diameter: Approx. 24mm / 0.95"
* Pump Length: Approx. 45mm / 1.8"
* Pump Height: Approx. 30mm / 1.2"
* Wire Length: ~13mm cm

**2.6.2 APPLICATIONS**

* Great for building science projects, fire-extinguishers, fire fighting robots, fountains, waterfalls, plant watering systems etc.
* Controlled fountain water flow
* Controlled Garden watering systems
* Hydroponic Systems
* Fresh water intake or exhaust systems for fish aquariums

**2.7 DC GEAR MOTOR ( 100RPM )**

100RPM Centre Shaft Economy Series DC Motor is high quality low cost DC geared motor. It has steel gears and pinions to ensure longer life and better wear and tear properties. The gears are fixed on hardened steel spindles polished to a mirror finish. The output shaft rotates in a plastic bushing. The whole assembly is covered with a plastic ring. Gearbox is sealed and lubricated with lithium grease and require no maintenance. The motor is screwed to the gear box from inside.  
Although motor gives 100 RPM at 12V but motor runs smoothly from 4V to 12V and gives wide range of RPM, and torque.

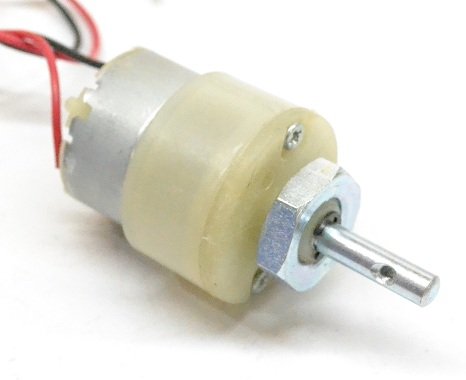


Fig. 2.9 : DC Gear Motor of 100rpm

**2.7.1 SPECIFICATIONS**

* DC supply: 4 to 12V
* RPM: 100 at 12V
* Total length: 46mm
* Motor diameter: 36mm
* Motor length: 25mm
* Brush type: Precious metal
* Gear head diameter: 37mm
* Gear head length: 21mm
* Output shaft: Centred
* Motor weight: 100gms

**2.8 VOLTAGE REGULATOR IC**

Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 IC, a member of 78xx series of fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC). The xx in 78xx indicates the output voltage it provides. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink.

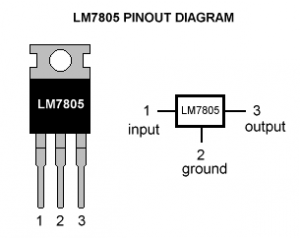


Fig. 2.10 : Voltage Regulator IC

# 2.8.1 7805 IC RATING

* Input voltage range 7V- 35V
* Current rating Ic =1A
* Output voltage range   VMax=5.2V ,VMin=4.8V

# 2.8.2 PIN DETAILS OF 7805 IC

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin No.** | **Pin** | **Function** | **Description** |
| 1 | INPUT | Input voltage (7V-35V) | In this pin of the IC positive unregulated voltage is given in regulation. |
| 2 | GROUND | Ground (0V) | In this pin where the ground is given. This pin is neutral for equally the input and output. |
| 3 | OUTPUT | Regulated output; 5V (4.8V-5.2V) | The output of the regulated 5V volt is taken out at this pin of the IC regulator. |

**2.9 SPST SWITCH**

A Single Pole Single Throw (SPST) switch is a switch that only has a single input and can connect only to one output. This means it only has one input terminal and only one output terminal.



Fig. 2.11 : SPST Switch

The Single Pole Single through (SPST) is a basic on/off switch that just connects or breaks the connection between two terminals. The [power supply](https://www.edgefxkits.com/auto-power-supply-control-from-4-different-sources-solar-mains-generator-inverter-to-ensure-no-break-power) to a circuit is switched by the SPST switch. A simple SPST switch is shown in figure below.

These types of switches are also called toggle switches. This switch has two contacts one is input and other output. From the typical light switch diagram, it controls one wire (pole) and it makes one connection (throw). This is an on/off switch, when the switch is closed or on then current flows through the terminals and the bulb in circuit will glow. When the switch is open or off then there is no current flow in the circuit.

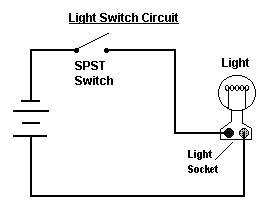
[](https://www.elprocus.com/wp-content/uploads/2013/07/SPST-Circuit.png)

Fig. 2.12 : SPST Circuit Diagram

**2.10 ARDUINO**

**Arduino Uno** is a microcontroller board based on 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header and a reset button.

 Out of these 14 pins, some pins have specific functions as listed below:

* **Serial Pins 0 (Rx) and 1 (Tx):** Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.
* **External Interrupt Pins 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.
* **PWM Pins 3, 5, 6, 9 and 11:**These pins provide an 8-bit PWM output by using analogWrite() function.
* **SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK):** These pins are used for SPI communication.
* **In-built LED Pin 13:** This pin is connected with an built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, its off.

Along with 14 Digital pins, there are 6 analog input pins, each of which provide 10 bits of resolution, i.e. 1024 different values. They measure from 0 to 5 volts but this limit can be increased by using AREF pin with analog Reference() function.

* Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.
* **Reset Pin:** Making this pin LOW, resets the microcontroller.
* **AREF:** Used to provide reference voltage for analog inputs with analogReference() function.

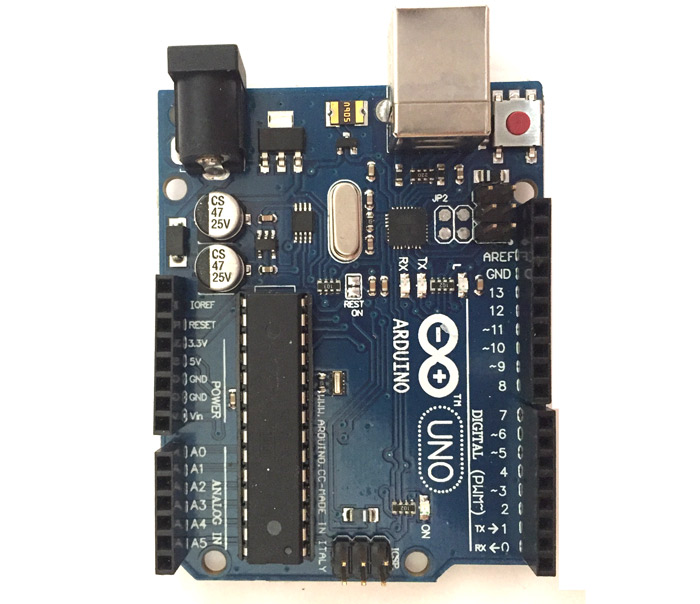


Fig. 2.13 : Arduino Uno

**2.10.1 APPLICATIONS**

* Prototyping of Electronics Products and Systems
* Multiple DIY Projects.
* Easy to use for beginner level DIYers and makers.
* Projects requiring Multiple I/O interfaces and communications.

**CHAPTER 3**

**SOFTWARE USED**

**3.1 ARDUINO IDE**

The Arduino IDE is incredibly minimalistic, yet it provides a near-complete environment for most Arduino-based projects. The top menu bar has the standard options, including “File” (new, load save, etc.), “Edit” (font, copy, paste, etc.), “Sketch” (for compiling and programming), “Tools” (useful options for testing projects), and “Help”. The middle section of the IDE is a simple text editor that where you can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program, and various other useful messages.

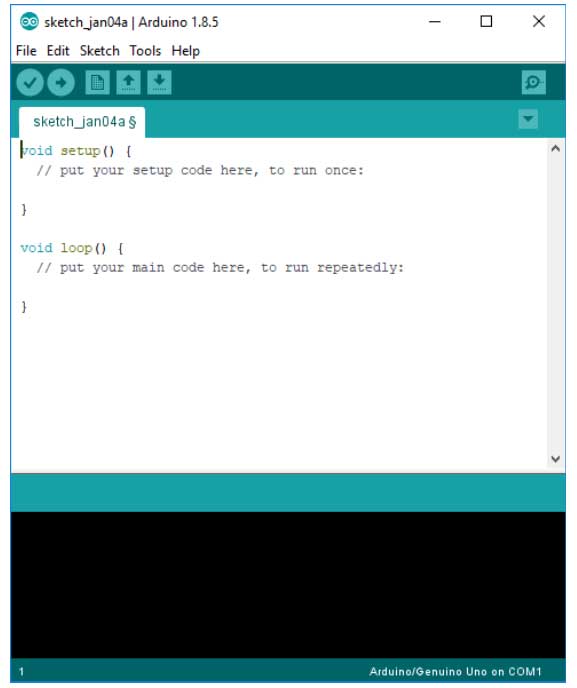


Fig. 3.1 : New File in Arduino IDE

Projects made using the Arduino are called sketches, and such sketches are usually written in a cut-down version of C++ (a number of C++ features are not included). Because programming a microcontroller is somewhat different from programming a computer, there are a number of device-specific libraries (e.g., changing pin modes, output data on pins, reading analog values, and timers). This sometimes confuses users who think Arduino is programmed in an “Arduino language.” However, the Arduino is, in fact, programmed in C++. It just uses unique libraries for the device.



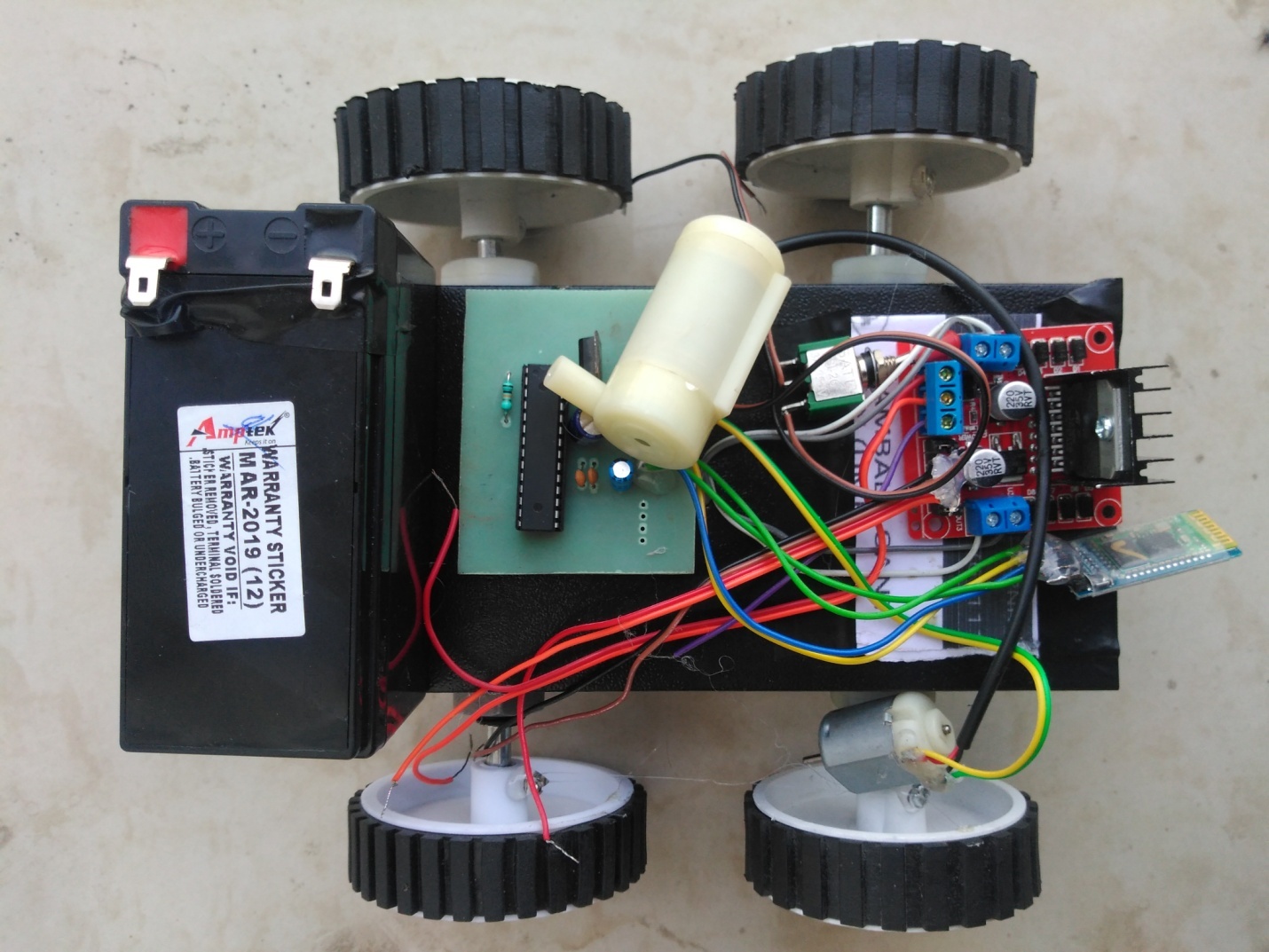
Fig. 3.2 : Arduino IDE Logo

**CHAPTER 4**

**RESULT AND ANALYSIS**

**5.1 INTRODUCTION**

This chapter presents the experimental results of the developed system prototype. The designed prototype of the robot is presented using screen shots. The circuit diagram was connected and it worked as expected, Figure below shows the developed Floor cleaning robot



**CONCLUSION**

A Cheaper and user friendly floor cleaner robot can be developed by using low cost semiconductor devices and ATMega32 . We can control the movement of the robot by using a android app which make it very easy to use and handle . Therefore it can be very economical for the people who cannot afford the high price of the cleaning robot . It can make our household chores more easy and of low maintenance .It can also use in the offices and government places.

**FUTURE SCOPE**

We can add much more functionality and features in the existing floor cleaning robot . It is very easy to make changes in the robot . Following updations are possible in the current prototype .

* Battery monitoring can be done by adding an LCD which can show us the battery percentage of the robot and whether it need charging to work or not.
* Self-charging of the robot will make it much more automatic and less dependent on human instructions
* Lighter body weight .
* Set alarm on/off time manually .

These future scopes of this project will increase the efficiency of this project.

**REFERENCE**

[1] https://circuitdigest.com/microcontroller-projects/arduino-floor-cleaning-robot

[2] https://www.instructables.com/id/CleanSweep-the-Floor-Cleaning-Robot/

[3]https://www.researchgate.net/publication/299372806\_Floor\_Cleaning\_Robot\_with\_Mobile-App\_or\_Autonomous

[4]https://create.arduino.cc/projecthub/theSTEMpedia/diy-floor-cleaning-robot-using-arduino-edb194

[5]http://www.standardsuniversity.org/wp-content/uploads/Smart-Floor-Cleaning-Robot-CLEAR.pdf

[6] <https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet>

**APPENDIX**

|  |
| --- |
| //MotorA |
|  | const int motorPin1 = 9; // Pin 14 of L293 |
|  | const int motorPin2 = 10; // Pin 10 of L293 |
|  | //Motor B |
|  | const int motorPin3 = 6; // Pin 7 of L293 |
|  | const int motorPin4 = 5; // Pin 2 of L293 |
|  |  |
|  | char data = 0; |
|  |  |
|  | void setup() |
|  | { |
|  | Serial.begin(9600); |
|  | pinMode(L2ED\_BUILTIN, OUTPUT); |
|  |  |
|  | //Set pins as outputs |
|  | pinMode(motorPin1, OUTPUT); |
|  | pinMode(motorPin2, OUTPUT); |
|  | pinMode(motorPin3, OUTPUT); |
|  | pinMode(motorPin4, OUTPUT); |
|  |  |
|  | //Motor Control - Motor A: motorPin1,motorpin2 & Motor B: motorpin3,motorpin4 |
|  |  |
|  | //This code will turn Motor A clockwise for 2 sec. |
|  | analogWrite(motorPin1, 180); |
|  | analogWrite(motorPin2, 0); |
|  | analogWrite(motorPin3, 180); |
|  | analogWrite(motorPin4, 0); |
|  | delay(500); |
|  | //This code will turn Motor A counter-clockwise for 2 sec. |
|  | analogWrite(motorPin1, 0); |
|  | analogWrite(motorPin2, 180); |
|  | analogWrite(motorPin3, 0); |
|  | analogWrite(motorPin4, 180); |
|  | delay(500); |
|  |  |
|  | //This code will turn Motor B clockwise for 2 sec. |
|  | analogWrite(motorPin1, 0); |
|  | analogWrite(motorPin2, 180); |
|  | analogWrite(motorPin3, 180); |
|  | analogWrite(motorPin4, 0); |
|  | delay(100); |
|  | //This code will turn Motor B counter-clockwise for 2 sec. |
|  | analogWrite(motorPin1, 180); |
|  | analogWrite(motorPin2, 0); |
|  | analogWrite(motorPin3, 0); |
|  | analogWrite(motorPin4, 180); |
|  | delay(100); |
|  |  |
|  | //And this code will stop motors |
|  | analogWrite(motorPin1, 0); |
|  | analogWrite(motorPin2, 0); |
|  | analogWrite(motorPin3, 0); |
|  | analogWrite(motorPin4, 0); |
|  |  |
|  | } |
|  |  |
|  |  |
|  | void loop() |
|  | { |
|  | if(Serial.available() > 0) // Send data only when you receive data: |
|  | { |
|  | data = Serial.read(); //Read the incoming data and store it into variable data |
|  | Serial.print(data); //Print Value inside data in Serial monitor |
|  | Serial.print("\n"); //New line |
|  | if(data == '1') // Checks whether value of data is equal to 1 |
|  | digitalWrite(LED\_BUILTIN, HIGH); //If value is 1 then LED turns ON |
|  | else if(data == '0') // Checks whether value of data is equal to 0 |
|  | digitalWrite(LED\_BUILTIN, LOW);; //If value is 0 then LED turns OFF |
|  | } |
|  |  |
|  |  |
|  | } |