MOBILE PHONE CONTROLLED ROBOT CAR USING G-SENSOR AND ARDUINO

Submitted by

RUTHRA MOORTHI K

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

TABLE OF CONTENTS

ABSTRACT

In this article, we are going to Control the Robot Car through the G sensor of our mobile phone and you will be able to move the Robot just by tilting the Phone. We will also use Arduino and RemoteXY app for this G-Sensor Controlled Robot. RemoteXY app is used to create the interface in the Smart Phone for controlling the Robot. We will add the joystick in the interface so that Robot can also be controlled by Joystick as well as by tilting the phone.

G-Sensor or Gravity sensor is basically Accelerometer in Smart phone which is used to control the screen orientation of the phone. Accelerometer senses the X,Y, Z directions of the Gravitational force and rotate the Screen according to alignment of the Phone. Now days, more sensitive and accurate Gyroscope sensor is used in mobiles for deciding the orientation of the Screen. In our Project, Robot car will move, according to the direction in which phone is being tilted, like when we tilt the phone forward, then car will move forward and we tilt it down then car will move backward. This is same like when we play some car games in Mobile, they also use G sensor to move the car accordingly

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In the recent days, the process of being developing is drastically improved, particularly increased in the communication fields such as Bluetooth, other remote-controlled cars and robots. In every country technology is currently developing with many users, especially in India every people are using different operating systems which are available in several smart phones. In olden days the communication between devices should be transmitted in wired and risky way without seeking any help of any person. But in present world the communication between devices is improved in wireless manner without risk and fast are kept feasible manner with taking help of persons. Android operating system is used to communicate between Smartphone hardware and several mobile applications. This designed model is controlled with the help of Bluetooth using Smartphone.

This robot is shaped like a car structure which contains of four wheels. This proposed designed is mainly used in different areas and fields in many industries Such as travelling goods from one place to another place and also moving many tiny particles in a fast manner. G-Sensor or Gravity sensor is basically Accelerometer in Smart phone which is used to control the screen orientation of the phone. Accelerometer senses the X,Y, Z directions of the Gravitational force and rotate the Screen according to alignment of the Phone.

Now days, more sensitive and accurate Gyroscope sensor is used in mobiles for deciding the orientation of the Screen. In our Project, Robot car will move, according to the direction in which phone is being tilted, like when we tilt the phone forward, then car will move forward and we tilt it down then car will move backward.

Being able to achieve reliable communication is an important open area of research to robotics as well as other technology areas. As an interest in robotics continues to grow, robots are increasingly being integrated in everyday life.

The results of this integration are end-users possessing less and less technical knowledge of the technology. A robot is basically an electrical or mechanical or electromechanical, programmable or non programmable multifunctional manipulator designed to move material, parts, tools or specialized devices through various programmed motions for the performance of a variety of tasks. Robots can be used to perform various tasks that are too dangerous or difficult for humans to implement directly. In this project, we aim at controlling a robot using tilt technique.

Although the appearance and capabilities of robots vary vastly, all robots share the feature of a mechanical, movable structure under some form of control. The control of robot involves three distinct phases: perception, processing and action. Generally, the preceptors are sensors mounted on the robot, processing is done by the on-board microcontroller or processor and the task is performed by using motors or some other actuators.

The project aims in designing a Robot car which is operated using bluetooth and also which is capable of Picking and Placing of many objects. The advent of latest high-speed technology and therefore the growing bluetooth capability provided realistic chance for brand spanking new automaton managements and realization of latest ways of control theory.

This technical improvement at the side of the requirement for top performance robots created quicker, more accurate and more intelligent robots using new robots control devices, new drivers and advanced control algorithms. This project describes a brand new economical resolution of automaton management systems. The bestowed automaton arm system are often used for various refined robotic applications. The modules in the project are: Bluetooth interfaced to Microcontroller, Robot car which is capable of Picking and placing objects, Servo motors is attached to the robot car for the movement of mechanism and Microcontroller that performs the dominant operations of mechanism arm in choosing and putting of objects. The controlling device of the whole system is a Microcontroller to which Bluetooth, Servo motors of robot car are interfaced through a motor driver. Whenever the appropriate keys on mobile application then the data will be transmitted through bluetooth to the microcontroller.

The Microcontroller checks the data with the program embedded in it and performs appropriate actions on the robot car. The Microcontroller is programmed using C language. An embedded system may be a combination of software package and hardware to perform a zealous task. Some of the most devices employed in embedded product square Measure Microprocessors and Microcontrollers. Microprocessors square measure remarked as general purpose processors as they merely settle for the inputs, process it and give the output. In contrast, a microcontroller not only accepts the data as inputs however additionally manipulates it, interfaces the information with numerous devices, controls the information and therefore finally offers the result.

The “Bluetooth controlled pick and place robot” using Arduino microcontroller is an exclusive project which is used to control speed and direction of Servo motor using general purpose bluetooth unremarkably.

CHAPTER 2

PROJECT DESCRIPTION

Control the Robot Car through the G sensor of our mobile phone and you will be able to move the Robot just by tilting the Phone. We will also use Arduino and RemoteXY app for this G-Sensor Controlled Robot. RemoteXY app is used to create the interface in the Smart Phone for controlling the Robot. We will add the joystick in the interface so that Robot can also be controlled by Joystick as well as by tilting the phone.

2.1 COMPONENTS REQUIRED

HARDWARE COMPONENTS:

- Two Wheel Robot Car Chassis
- Arduino UNO
- L298N Motor Driver
- HC05-Bluetooth Module
- 12v & 5v

SOFTWARE COMPONENTS:

- Arduino IDE
- RemoteXY Application

2.2 ARDUINO UNO

The Arduino UNO R3 is frequently used microcontroller board in the family of an Arduino. This is the latest third version of an Arduino board and released in the year 2011. The main advantage of this board is if we make a mistake we can change the microcontroller on the board. The main features of this board mainly include, it is available in DIP (dual-inline-package), detachable and ATmega328 microcontroller. The programming of this board can easily be loaded by using an Arduino computer program. This board has huge support from the Arduino community, which will make a very simple way to start working in embedded electronics, and many more applications. Please refer the link to know about Arduino – Basics, and Design

Arduino Uno R3 is one kind of ATmega328P based microcontroller board. It includes the whole thing required to hold up the microcontroller; just attach it to a PC with the help of a USB cable, and give the supply using AC-DC adapter or a battery to get started. The term Uno means “one” in the language of “Italian” and was selected for marking the release of Arduino IDE 1.0 software. The R3 Arduino Uno is the 3rd as well as most recent modification of the Arduino Uno. Arduino board and IDE software are the reference versions of Arduino and currently progressed to new releases. The Uno-board is the primary in a sequence of USB-Arduino boards, & the reference model designed for the Arduino platform.

FEATURES

- **Memory**
  - AVR CPU at up to 16 MHz
  - 32KB Flash
  - 2KB SRAM
  - 1KB EEPROM
  - Debug WIRE interface for on-chip debugging and programming
- **Security**
  - Power On Reset (POR)

- Brown Out Detection (BOD)

- **Peripherals**
    - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
    - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
    - 1x USART with fractional baud rate generator and start-of-frame detection
    - 1x controller/peripheral Serial Peripheral Interface (SPI)
    - 1x Dual mode controller/peripheral I2C
    - 1x Analog Comparator (AC) with a scalable reference input
    - Watchdog Timer with separate on-chip oscillator
    - Six PWM channels
    - Interrupt and wake-up on pin change
    - 8-bit AVR® RISC-based microcontroller

- **Power**

- The input voltage ranges from 6v to 20V
- 2.7-5.5 volts
- DC Current for each input/output pin is 40 mA
- DC Current for 3.3V Pin is 50 mA
- The recommended input voltage will range from 7v to 12V

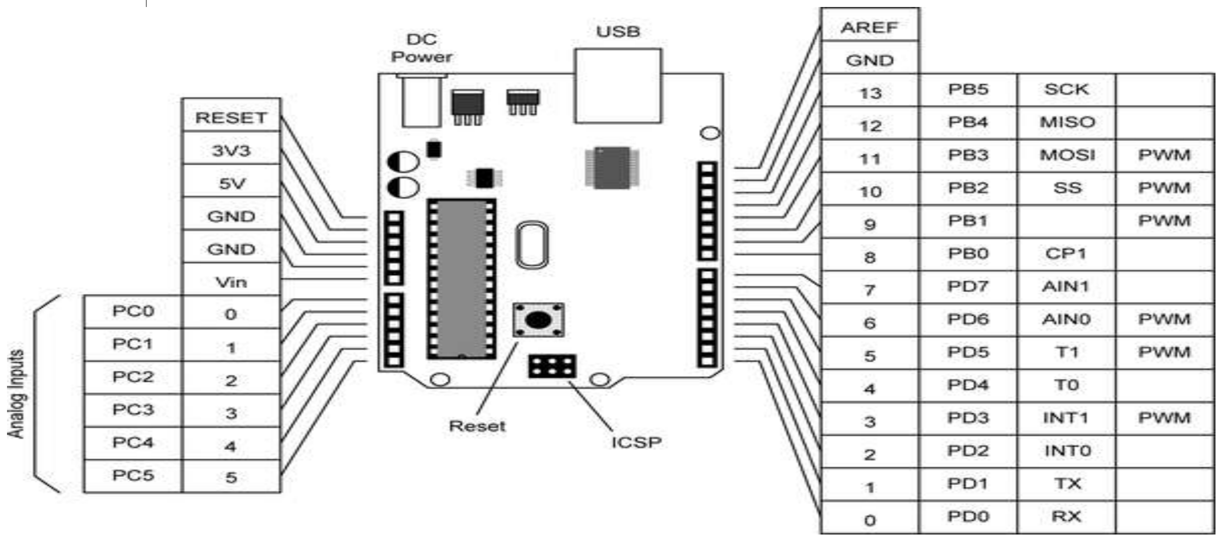

Fig1: PIN DESCRIPTION OF ARDUINO UNO

HARDWARE DETAILS:

- **Vin:** The input voltage or Vin to the Arduino while it is using an exterior power supply opposite to volts from the connection of USB or else RPS (regulated power supply). By using this pin, one can supply the voltage.

- **5Volts:** The RPS can be used to give the power supply to the microcontroller as well as components which are used on the Arduino board. This can approach from the input voltage through a regulator.

- **3V3:** A 3.3 supply voltage can be generated with the onboard regulator, and the highest draw current will be 50 mA.
- **GND:** Ground pins

- **Memory:** The memory of an ATmega328 microcontroller includes 32 KB and 0.5

KB memory is utilized for the Boot loader), and also it includes SRAM-2 KB as well as EEPROM-1KB.

- **Input and Output:** We know that an arguing Uno R3 includes 14-digital pins which can be used as an input otherwise output by using the functions like pin Mode (), digital Read(), and digital Write(). These pins can operate with 5V, and every digital pin can give or receive 20mA, & includes a 20k to 50k ohm pull up resistor. The maximum current on any pin is 40mA which cannot surpass for avoiding the microcontroller from the damage. Additionally, some of the pins of an Arduino include specific functions.

- **Serial Pins:** The serial pins of an Arduino board are TX (1) and RX (0) pins and these pins can be used to transfer the TTL serial data. The connection of these pins can be done with the equivalent pins of the ATmega8 U2 USB to TTL chip.

- **External Interrupt Pins:** The external interrupt pins of the board are 2 & 3, and these pins can be arranged to activate an interrupt on a rising otherwise falling edge, a low-value otherwise a modify in value

- **PWM Pins:** The PWM pins of an Arduino are 3, 5, 6, 9, 10, & 11, and gives an output of an 8-bit PWM with the function analog Write ().

- **SPI (Serial Peripheral Interface) Pins:** The SPI pins are 10, 11, 12, 13 namely SS, MOSI, MISO, SCK, and these will maintain the SPI communication with the help of the SPI library.

- **LED Pin:** An arguing board is inbuilt with a LED using digital pin-13. Whenever the digital pin is high, the LED will glow otherwise it will not glow.

- **TWI (2-Wire Interface) Pins:** The TWI pins are SDA or A4, & SCL or A5, which can support the communication of TWI with the help of Wire library.

- **AREF (Analog Reference) Pin:** An analog reference pin is the reference voltage to the inputs of an analog inputs using the function like analog reference().

- **Reset (RST) Pin:** This pin brings a low line for resetting the microcontroller, and it is very useful for using an RST button toward shields which can block the one over the Arduino R3 board.

- **Communication:** The communication protocols of an Arduino Uno include SPI, I2C, and UART serial communication.

- **UART:** An Arduino Uno uses the two functions like the transmitter digital pin1 and the receiver digital pin0. These pins are mainly used in UART TTL serial communication.

- **I2C:** An Arduino UNO board employs SDA pin otherwise A4 pin & A5 pin otherwise SCL pin is used for I2C communication with wire library. In this, both the SCL and SDA are CLK signal and data signal.

- **SPI Pins:** The SPI communication includes MOSI, MISO, and SCK.

- **MOSI (Pin11):** This is the master out slave in the pin, used to transmit the data to the devices

- **MISO (Pin12):** This pin is a serial CLK, and the CLK pulse will synchronize the transmission of which is produced by the master.

- **SCK (Pin13):** The CLK pulse synchronizes data transmission that is generated by the master. Equivalent pins with the SPI library is employed for the communication of SPI. ICSP (in-circuit serial programming) headers can be utilized for programming ATmega microcontroller directly with the boot loader.

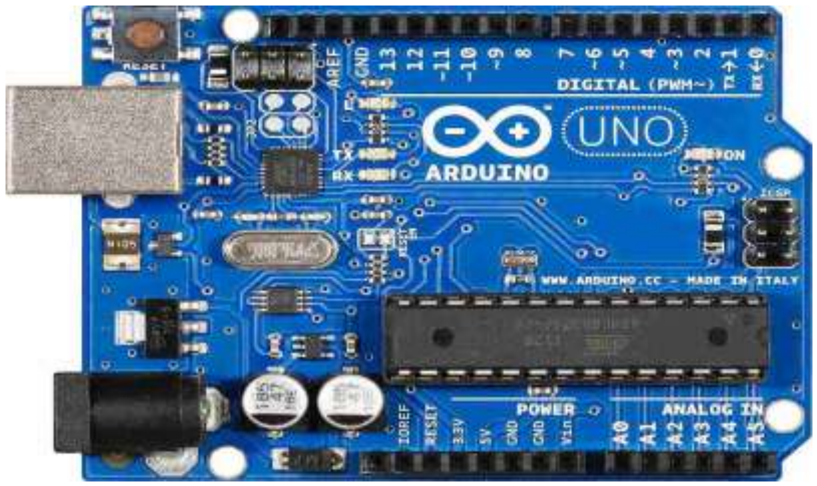

Fig2: **ARDUINO UNO DEVELOPMENT BOARD**

SOFTWARE DETAILS:

- The programming of an Arduino Uno R3 can be done using IDE software. The microcontroller on the board will come with pre-burned by a boot loader that permits to upload fresh code without using an exterior hardware programmer.
- The communication of this can be done using a protocol like STK500.
- We can also upload the program in the microcontroller by avoiding the boot loader using the header like the In-Circuit Serial Programming.

2.3 ARDUINO IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment.

The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'

Programs written using Arduino Software (IDE) are called sketches. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

FILE:

- New Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- Open Allows to load a sketch file browsing through the computer drives and folders.
- Open Recent Provides a short list of the most recent sketches, ready to be opened.
- Sketchbook Shows the current sketches within the sketchbook folder structure.

- Clicking on any name opens the corresponding sketch in a new editor instance.
- Examples Any example provided by the Arduino Software (IDE) or library shows up in this menu item.
- All the examples are structured in a tree that allows easy access by topic or library.
- Close Closes the instance of the Arduino Software from which it is clicked.
- Save Saves the sketch with the current name.
- If the file hasn't been named before, a name will be provided in a "Save as.." window.
- Save as... Allows to save the current sketch with a different name.
- Page Setup It shows the Page Setup window for printing.
- Print Sends the current sketch to the printer according to the settings defined in Page Setup.
- Preferences Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
- Quit Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

EDIT:

- Undo/Redo Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- Cut Removes the selected text from the editor and places it into the clipboard.
- Copy Duplicates the selected text in the editor and places it into the clipboard.
- Copy for Forum Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.

- Copy as HTML Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- Paste Puts the contents of the clipboard at the cursor position, in the editor.
- Select All Selects and highlights the whole content of the editor.
- Comment/Uncomment Puts or removes the // comment marker at the beginning of each selected line.
- Increase/Decrease Indent Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- Find Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- Find Next Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.
- Find Previous Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

SKETCH:

- Verify/Compile Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.
- Upload Compiles and loads the binary file onto the configured board through the configured Port.
- Upload Using Programmer This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Boot loader command must be executed.

- Export Compiled Binary Saves a .hex file that may be kept as archive or sent to the board using other tools.
- Show Sketch Folder Opens the current sketch folder.
- Include Library Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.
- Add File... Adds a supplemental file to the sketch (it will be copied from its current location). The file is saved to the subfolder of the sketch, which is intended for assets such as documentation. The contents of the folder are not compiled, so they do not become part of the sketch program.

HELP:

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

Find in Reference This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

LIBRARIES:

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

UPLOAD:

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an UNO or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

BOARDS:

The board selection has two effects: it sets the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and sets and the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the bootloader. You can find different boards here.

Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. The Boards Manager included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

- Arduino Yún An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- Arduino Uno An ATmega328P running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Diecimila or Duemilanove w/ ATmega168 An ATmega168 running at 16 MHz with auto-reset.
- Arduino Nano w/ ATmega328P An ATmega328P running at 16 MHz with auto-reset. Has eight analog inputs.
- Arduino Mega 2560 An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.

- Arduino Mega An ATmega1280 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- Arduino Mega ADK An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- Arduino Leonardo An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- Arduino Micro An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- Arduino Esplora An ATmega32u4 running at 16 MHz with auto-reset.
- Arduino Mini w/ ATmega328P An ATmega328P running at 16 MHz with auto-reset, 8 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Ethernet Equivalent to Arduino UNO with an Ethernet shield: An ATmega328P running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Fio An ATmega328P running at 8 MHz with auto-reset. Equivalent to Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328P, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino BT w/ ATmega328P ATmega328P running at 16 MHz. The bootloader burned (4 KB) includes codes to initialize the on-board Bluetooth® module, 6 Analog In, 14 Digital I/O and 6 PWM..
- LilyPad Arduino USB An ATmega32u4 running at 8 MHz with auto-reset, 4 Analog In, 9 Digital I/O and 4 PWM.
- LilyPad Arduino An ATmega168 or ATmega132 running at 8 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328P An ATmega328P running at 16 MHz with auto-reset. Equivalent to Arduino Duemilanove or Nano

w/ ATmega328P; 6 Analog In, 14 Digital I/O and 6 PWM.

- Arduino NG or older w/ ATmega168 An ATmega168 running at 16 MHzwithout auto-reset. Compilation and upload is equivalent to Arduino Diecimila or Duemilanove w/ ATmega168, but the bootloader burned has a slower timeout (and blinks the pin 13 LED three times on reset); 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Robot Control An ATmega328P running at 16 MHz with auto-reset.
- Arduino Robot Motor An ATmega328P running at 16 MHz with auto-reset.
- Arduino Gemma An ATtiny85 running at 8 MHz with auto-reset, 1 Analog In, 3 Digital I/O and 2 PWM.

SERIAL MONITOR:

This displays serial sent from the Arduino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

The Serial Plotter tool is a versatile tool for tracking different data that is sent from your Arduino board. It functions similarly to your standard Serial Monitor tool which is used to print data "terminal style", but is a greater visual tool that will help you understand and compare your data better.

2.4 L298N MOTOR DRIVER

L298N IC:

The L298 is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

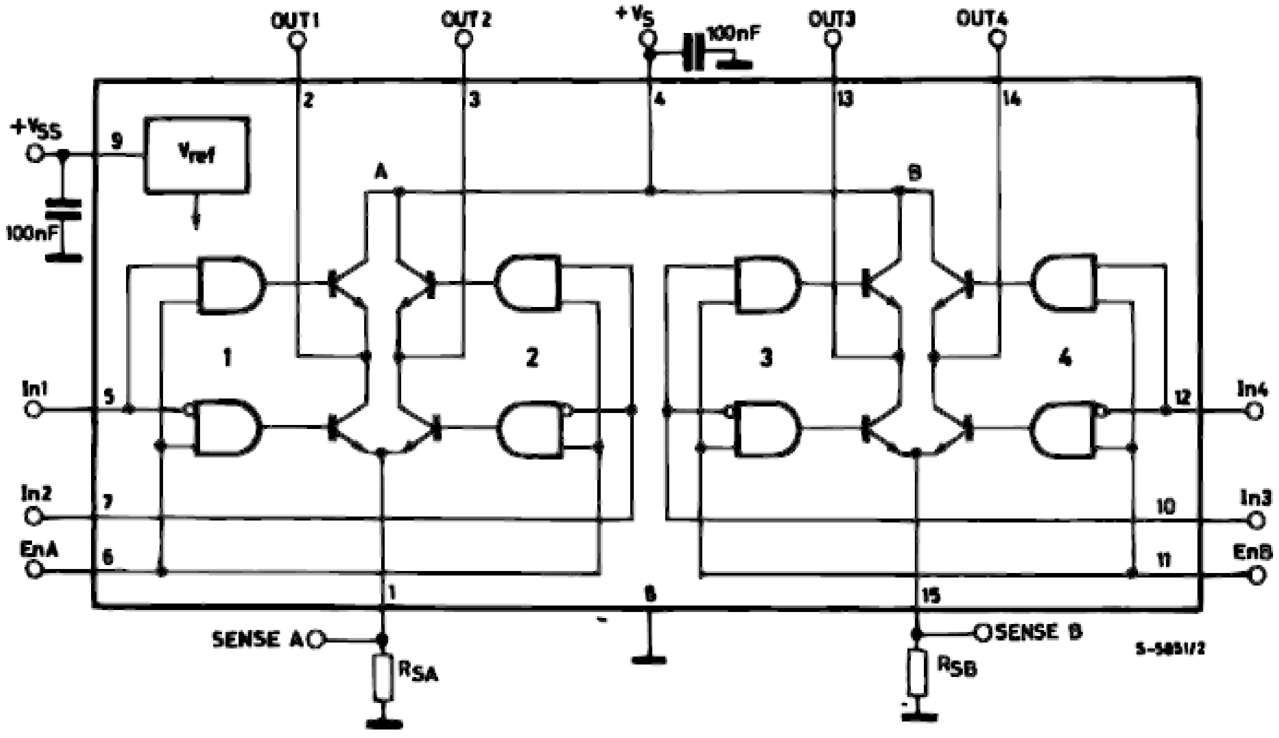

Fig3: **L298N BLOCK DIAGRAM**

PIN DETAILS OF L298N IC:

Table1: L298N PIN DETAILS

L298N IC pins	**Name**	**Function**
1,15	Sense A, Sense B	Between this pin and the ground, a sense resistor is connected to control the current of the load.
2,3	Output 1, Output 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	VS	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5,7	Input 1, Input 2	TTL Compatible Inputs of the Bridge A.
6,11	Enable A, Enable B	TTL Compatible Enable Input: the L state disables the bridge A(enable A) and/or the bridge B (enable B).
8	GND	Ground
9	VSS	Supply Voltage for the Logic Blocks. (A100nF capacitor must be connected between this pin and ground.)
10,12	Input 3, Input 4	TTL Compatible Inputs of the Bridge B.
13,14	Output 3, Output 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at the pin.

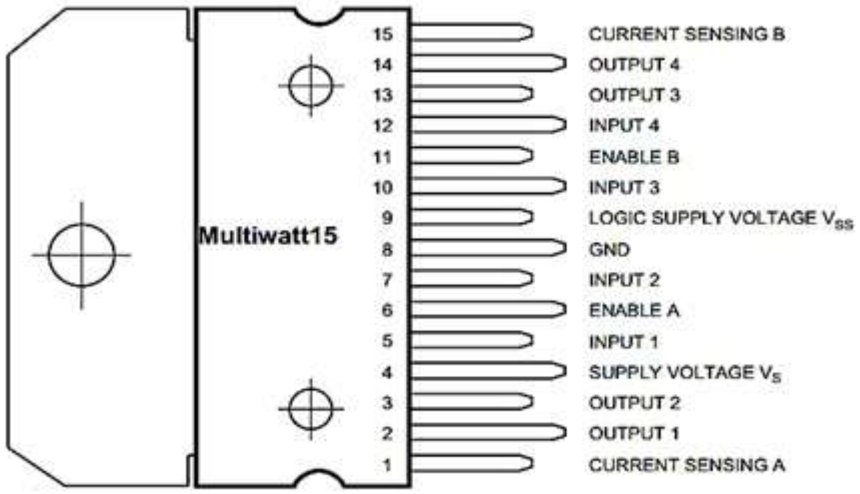

Fig4: L298N IC PIN DIAGRAM

L298N MODULE:

This L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control.

The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit.

78M05 Voltage regulator will be enabled only when the jumper is placed. When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller. The jumper should not be placed when the power supply is greater than 12V and separate 5V should be given through 5V terminal to power the internal circuitry.

ENA & ENB pins are speed control pins for Motor A and Motor B while IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and Motor B.

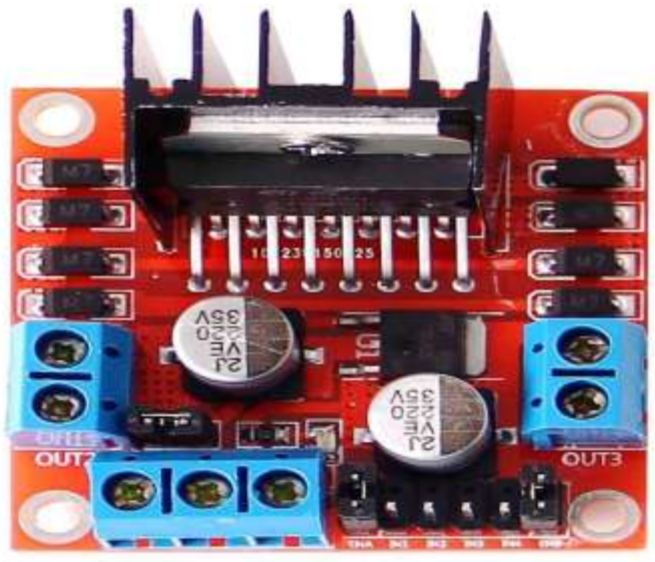

Fig5: **L298N MODULE**

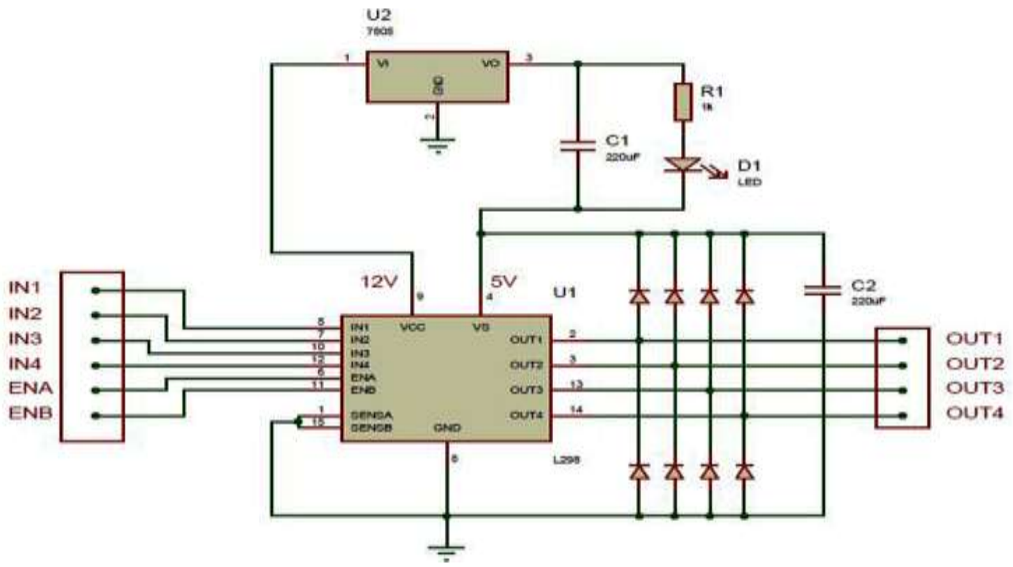

Fig6: **L298N MODULE BLOCK DIAGRAM**

L298N MODULE PIN DETAILS:

Table2: L298N MODULE PIN DETAILS

Pin Name	Description
IN1, IN2	Motor A input pins. Used to control the spinning direction of Motor A
IN3, IN4	Motor B input pins. Used to control the spinning direction of Motor B
ENA	Enables PWM signal for Motor A
ENB	Enables PWM signal for Motor B
OUT1, OUT2	Output pins of Motor A
OUT3, OUT4	Output pins of Motor B
12V	12V input from DC power Source
5V	Supplies power for the switching logic circuitry inside L298N IC
GND	Ground pin

FEATURES & SPECIFICATIONS:

- Driver Model: L298N 2A
- Driver Chip: Double H Bridge L298N
- Motor Supply Voltage (Maximum): 46V

- Motor Supply Current (Maximum): 2A
- Logic Voltage: 5V
- Driver Voltage: 5-35V
- Driver Current:2A
- Logical Current:0-36mA
- Maximum Power (W): 25W
- Current Sense for each motor
- Heatsink for better performance
- Power-On LED indicator

2.5 HC-05 BLUETOOTH MODULE:

- HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.

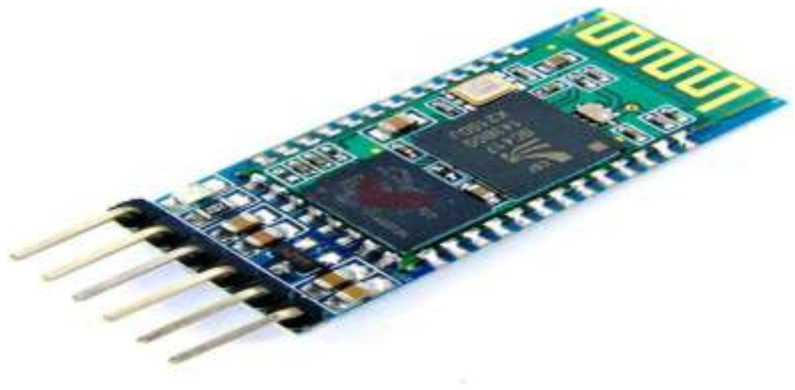

Fig7: **HC-05 BLUETOOTH MODULE**

The HC05 Bluetooth module is used as UART serial converter module and can easily transfer the UART data through the wireless Bluetooth. The Bluetooth module has a Frequency: 2.4GHz ISM band, PIO control and comes with an integrated antenna and edge connector.

HC-05 BLUETOOTH MODULE PIN DETAILS:

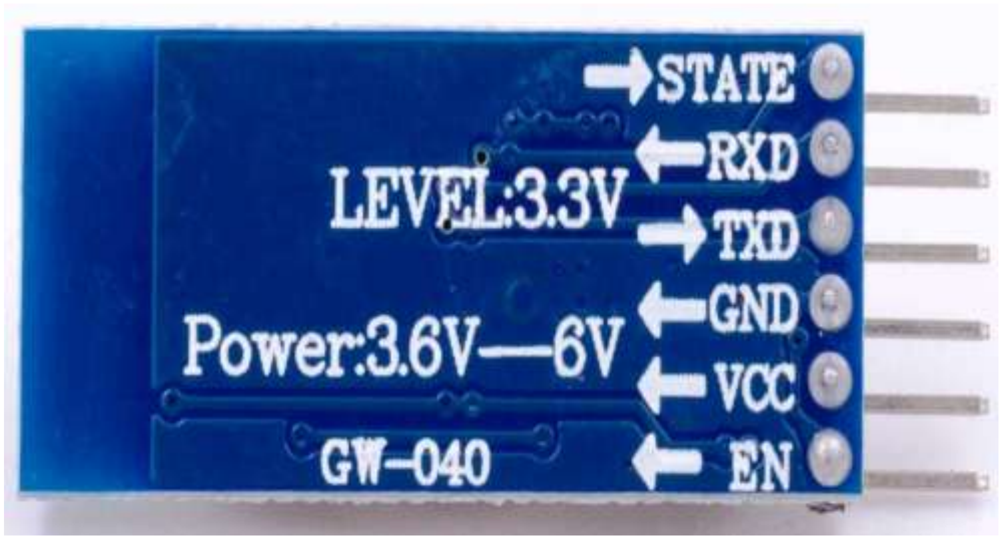

Fig8: **HC05-BLUETOOTH MODULE PIN DIAGRAM**

Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth.

It has 6 pins,

- **Key/EN:** It is used to bring Bluetooth module in AT commands mode. If Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode.

HC-05 module has two modes,

- **Data mode:** Exchange of data between devices.
- **Command mode:** It uses AT commands which are used to change setting of HC-05. To send these commands to module serial (USART) port is used.

- **VCC:** Connect 5 V or 3.3 V to this Pin.
- **GND:** Ground Pin of module.

- **TXD:** Transmit Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)
- **RXD:** Receive data serially (received data will be transmitted wirelessly by Bluetooth module).
- **State:** It tells whether module is connected or not.

HC-05 MODULE INFORMATION:

- HC-05 has red LED which indicates connection status, whether the Bluetooth is connected or not. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds.
- This module works on 3.3V. We can connect 5V supply voltage as well since the module has on board 5 to 3.3 V regulator.
- As HC-05 Bluetooth module has 3.3V level for RX/TX and microcontroller can detect 3.3 V level, so, no need to shift transmit level of HC-05 module. But we need to shift the transmit voltage level from microcontroller to RX of HC-05 module.
- The data transfer rate of HC-05 module can vary up to 1Mbps is in the range of 10 meters.

SPECIFICATION:

- Bluetooth version: 2.0 + EDR (Enhanced Data Rate)
- Frequency: 2.4 GHz ISM band
- Modulation: GFSK (Gaussian Frequency Shift Keying)

- Transmit power: Class 2 (up to 4 dBm)
- Sensitivity: -80 dBm typical
- Range: approximately 10 meters (or 33 feet) in open air
- Profiles supported: SPP (Serial Port Profile), HID (Human Interface Device) and others
- Operating voltage: 3.3V to 5V DC
- Operating current: less than 50mA
- Standby current: less than 2.5mA
- Sleep current: less than 1mA
- Interface: UART (Universal Asynchronous Receiver/Transmitter)
- Baud rates: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, and 460800
- Operating temperature: -20°C to 75°C (-4°F to 167°F)\

BLUETOOTH COMMUNICATION BETWEEN DEVICES:

E.g. Send data from Smartphone terminal to HC-05 Bluetooth module and see this data on PC serial terminal and vice versa.

To communicate smartphone with HC-05 Bluetooth module, smartphone requires Bluetooth terminal application for transmitting and receiving data. You can find Bluetooth terminal applications for android and windows in respective app. store.

So, when we want to communicate through smartphone with HC-05 Bluetooth module, connect this HC-05 module to the PC via serial to USB converter.

Before establishing communication between two Bluetooth devices, 1st we need to pair HC-05 module to smartphone for communication.

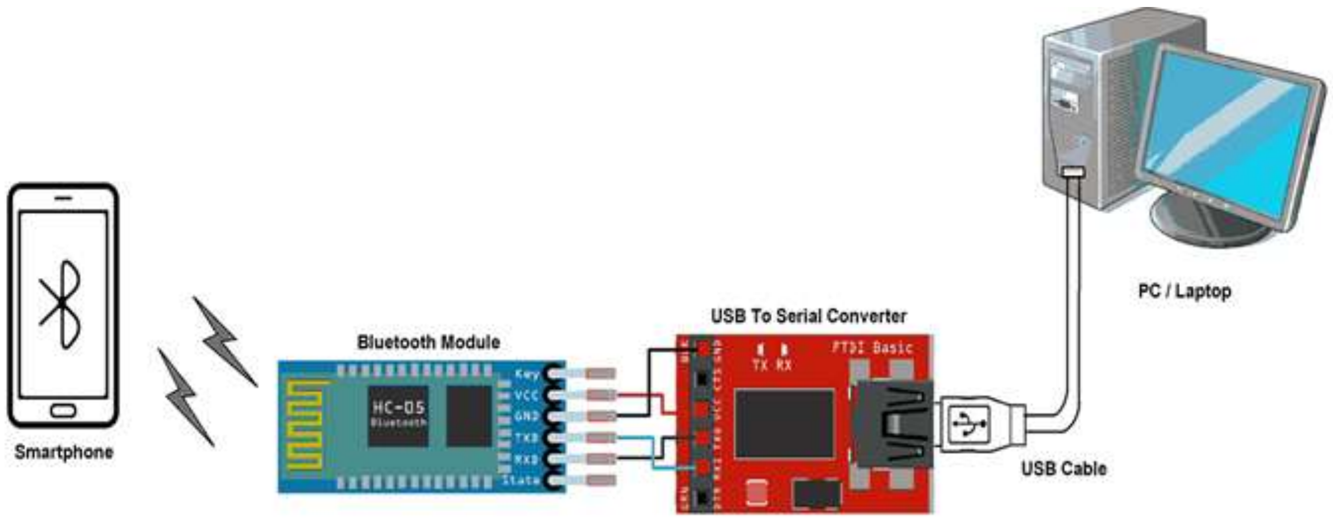

Fig9: **BLUETOOTH MODULE SERIAL INTERFACE**

PAIR HC-05 AND SMARTPHONE

- Search for new Bluetooth device from your phone. You will find Bluetooth device with “HC-05” name.
- Click on connect/pair device option; default pin for HC-05 is 1234 or 0000.

After pairing two Bluetooth devices, open terminal software (e.g. Teraterm, Realterm etc.) in PC, and select the port where we have connected USB to serial module. Also select default baud rate of 9600 bps.

In smart phone, open Bluetooth terminal application and connect to paired device HC-05.

It is simple to communicate, we just have to type in the Bluetooth terminal application of smartphone. Characters will get sent wirelessly to Bluetooth module HC-05. HC-05 will automatically transmit it serially to the PC, which will appear on terminal. Same way we can send data from PC to smartphone.

COMMAND MODE:

- When we want to change settings of HC-05 Bluetooth module like change password for connection, baud rate, Bluetooth device's name etc.
- To do this, HC-05 has AT commands.
- To use HC-05 Bluetooth module in AT command mode, connect "Key" pin to High (VCC).

Table3: HC05-BLUETOOTH COMMANDS

Command	Description	Response
AT	Checking communication	OK
AT+PSWD=XXXX	Set Password e.g. AT+PSWD=4567	OK
AT+NAME=XXXX	Set Bluetooth Device Name e.g. AT+NAME=MyHC-05	OK
AT+UART=Baud rate, stop bit, parity bit	Change Baud rate e.g. AT+UART=9600,1,0	OK
AT+VERSION?	Respond version no. of Bluetooth module	+Version: XX OK e.g. +Version: 2.0 20130107 OK
AT+ORGL	Send detail of setting done by manufacturer	Parameters: device type, module mode, serial parameter, passkey, etc.

- Default Baud rate of HC-05 in command mode is 38400bps.

- Following are some AT command generally used to change setting of Bluetooth module.
- To send these commands, we have to connect HC-05 Bluetooth module to the PC via serial to USB converter and transmit these command through serial terminal of PC.

ALTERNATE OPTIONS FOR HC-05 BLUETOOTH MODULE:

- **HC-06 Bluetooth module**: This is a similar module to the HC-05, but it is limited to a slave role only. It has a smaller form factor and is generally cheaper than the HC-05. However, it does not support some of the advanced features of the HC-05, such as the ability to enter AT mode to configure the module.

- **HM-10 Bluetooth module**: This is a more advanced Bluetooth module that supports Bluetooth 4.0 (BLE) and can act as both a master and slave device. It also supports a wider range of AT commands for configuring the module, and has a longer range than the HC-05. However, it is generally more expensive than the HC-05.

- **RN-42 Bluetooth module**: This is another Bluetooth module that supports both the SPP and HID profiles, similar to the HC-05. It has a longer range than the HC-05 and supports faster data rates. However, it is also more expensive and may require additional configuration to work properly.

- **ESP32 Bluetooth module**: This is a powerful Wi-Fi and Bluetooth module that includes a dual-core processor and support for both Bluetooth Classic and BLE. It is more expensive than the HC-05, but offers more advanced features and capabilities.

- **nRF24L01+ Wireless module**: This is a wireless module that operates at 2.4GHz and uses a different protocol than Bluetooth. It is generally cheaper than Bluetooth modules and can be used for applications where a shorter range and lower data rate are acceptable.

2.6 REMOTEXY APPLICATION

RemoteXY is easy way to make and use a mobile graphical user interface for controller boards to control via smartphone or tablet.

The system includes:

- Editor of mobile graphical interfaces for controller boards, located on the site remotexy.com
- Mobile app RemoteXY that allows to connect to the controller and control it via graphical interface.

DISTINCTIVE FEATURES:

- The interface structure is stored in the controller. When connected, there is no interaction with servers to download the interface. The interface structure is downloaded to the mobile application from the controller.
- One mobile application can manage all your devices. The number of devices is not limited.

SUPPORTED CONNECTION METHODS:

- Internet over Cloud Server
- WiFi client and access point
- Bluetooth
- Ethernet by IP or URL
- USB OTG (Android only that support USB OTG)
- Supported boards

- Arduino UNO, MEGA, Leonardo, Pro Mini, Nano, MICRO and compatible AVR boards
- ESP8266 boards
- ESP32 boards
- STM32F1 boards
- ChipKIT UNO32, ChipKIT uC32, ChipKIT Max32

SUPPORTED CONNECTION MODULES:

- Bluetooth HC-05, HC-06 or compatible
- Bluetooth BLE HM-10 or compatible
- ESP8266 as modem
- Ethernet W5100

SUPPORTED IDE:

- Arduino IDE
- FLProg IDE
- MPIDE

SUPPORTED MOBILE OS:

- Android
- iOS

RemoteXY is easy way to make a unique graphical interface to control microcontroller device via mobile application, Arduino for example.

REMOTEXY ALLOWS:

- To develop any graphical management interface, using the control, display and decoration elements any combination thereof. You can develop the graphical interface for any task, placing the elements on the screen using the online editor. Online editor posted on the website remotexy.com.

- After the development of the graphical interface, you get the source code for the microcontroller that implements your interface.
- The source code provides a structure for interaction between your program with the controls and display.
- Thus you can easily integrate the control system into your task for which you are developing the device.
- To manage microcontroller device using your smartphone or tablet with the graphical interface. For manage used mobile application RemoteXY.
- Using one mobile application, you can manage a large number of devices with different graphical management interfaces.
- As the interface description is stored on board the microcontroller device.
- It is easy way to make a unique graphical interface to control microcontroller device via mobile application, Arduino for example.
- To develop any graphical management interface, using the control, display and decoration elements any combination there of.
- To use the slider as a speed control we need to connect it to PIN 6 that controls the NPN transistor. We need to add two lines of code to connect the RemoteXY slider to PIN 6. We add them to the loop function. int MotorSpeed = RemoteXY.

CHAPTER 3
WORKING PRINCIPLE

First of all, we will interface the L298N motor controller with the Arduino. Connect the ENA and ENB pin of the motor controller to the Arduino pin 12 and 11 respectively. These two pins are for the PWM control of the motor. Using theses pins, we can increase or decrease the speed of car. Then connect the IN1, IN2, IN3 and IN4 to the Arduino pins 10, 9, 8 and 7 respectively. These pins will rotate the motors in both directions (clockwise and anti-clockwise).

To power the motor, connect the positive and negative of the battery to the 12V and the ground of the motor controller. Then connect the 5V and the ground from the motor controller to the Arduino Vin and the ground.

Then we will connect the Bluetooth module HC-05 with the arduino. If you have HC-05, then it will work too. Connect the VCC and the ground of the Bluetooth module to the 5V and the ground of the Arduino. Then connect the TX pin of Bluetooth Module to the pin 2 of Arduino and the RX pin to the pin 3 of Arduino.

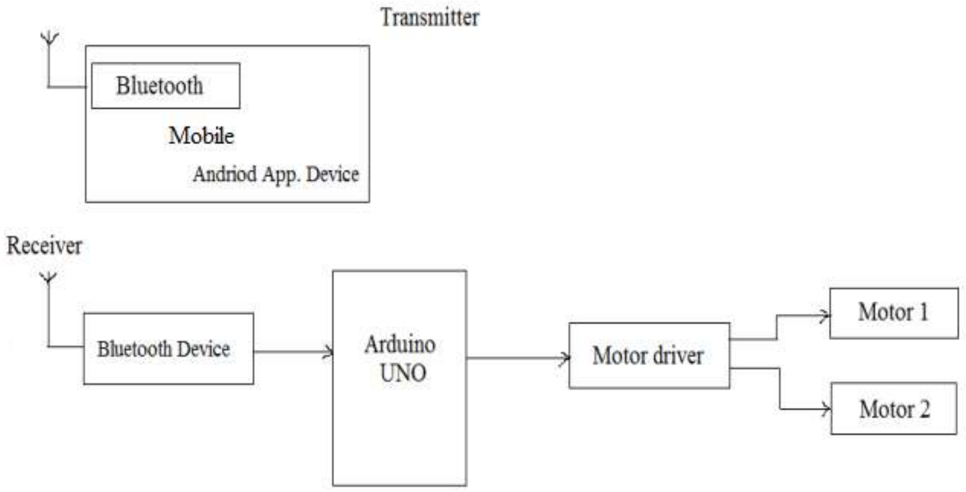

Fig10: BLOCK DIAGRAM OF MOBILE CONTROLLED ROBOT USING G-SENSOR AND ARDUINO

Bluetooth controlled car moves according to button touched in the android Bluetooth mobile app. To run this project first we need to download Bluetooth app form Google play store. We can use RemoteXY app that supporting or can send data.

When we tilt forward button in Bluetooth controller app then car start moving in forward direction and moving continues forward until next command comes.

When we tilt backward button in Bluetooth controller app then car start moving in reverse direction and moving continues reverse until next command comes.

When we tilt left button in Bluetooth controller app then car start moving in left direction and moving continues left until next command comes. In this condition front side motor turns front side wheels in left direction and rear motor runs in forward direction.

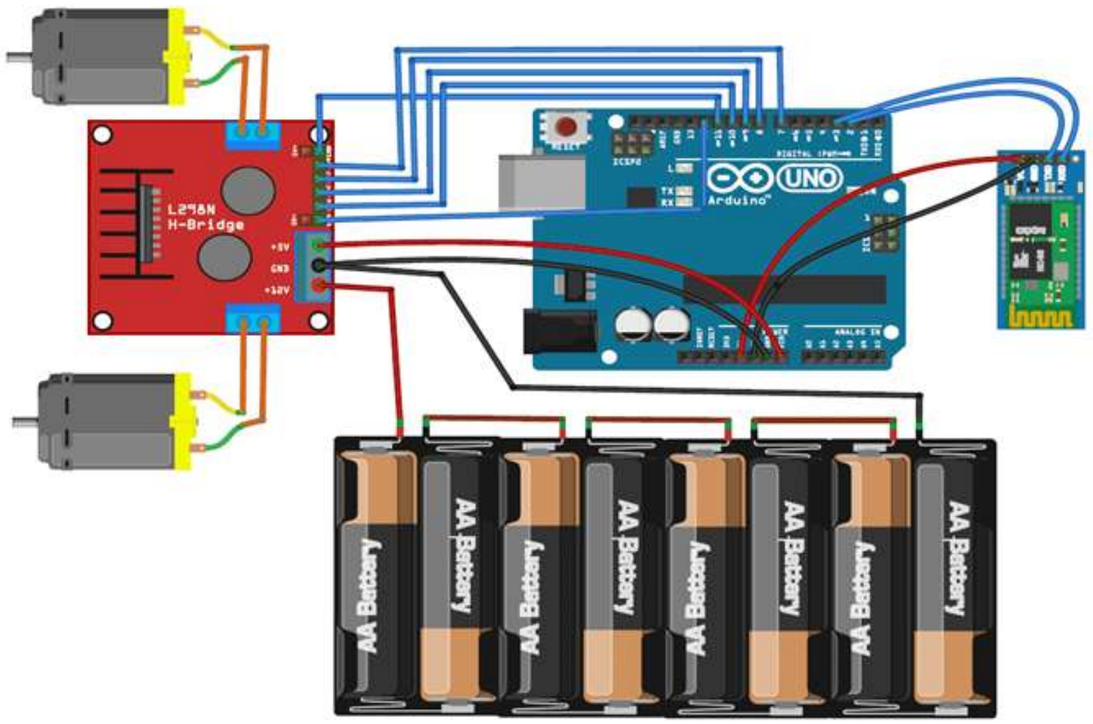

Fig11: CIRCUIT DIAGRAM OF MOBILE CONTROLLED ROBOT USING G-SENSOR AND ARDUINO

The microcontroller is pre programmed to take a decision for any given input and output sits decision to motor drivers in order to drive the motors for forward or backward motion or a turn.

The mobile that makes a call to the mobile phone stacked in the robot acts as a remote. So this simple robotic project does not require the construction of receiver and transmitter units when constructing any robot, one major mechanical constraint is the number there a two-wheel drive or a four-wheel drive.

Though four-wheel drive is more complex than two-wheel drive, it provides more torque and good control. Two-wheel drive, on the other hand, is very easy to construct. Motors are fixed to the bottom of this sheet and the circuit is affixed firmly on top of the sheet.

A cell phone is also mounted on the sheet as shown in the picture. In the four-wheel drive system, the two motors on a side are controlled in parallel. So a single L293D driver IC can drive the rover.

This model structure is merely the foundation of a fundamental execution. This sector provides an enormous amount of increase for potential work and study. The general structure has room for improvement, with the goal of being able to function and adapt in various requirements of circumstances. The module can be additionally reached out to acknowledge voice as a contribution to give guidelines to the designed vehicle.

CHAPTER 4

RESULTS AND DISCUSSIONS

This is how we can use the Gravity Sensor inside our Mobile phone to move the Robot. You can further experiment and find more interesting use of G sensor to control the outside things by interfacing a Microcontorller in between (like Arduino).

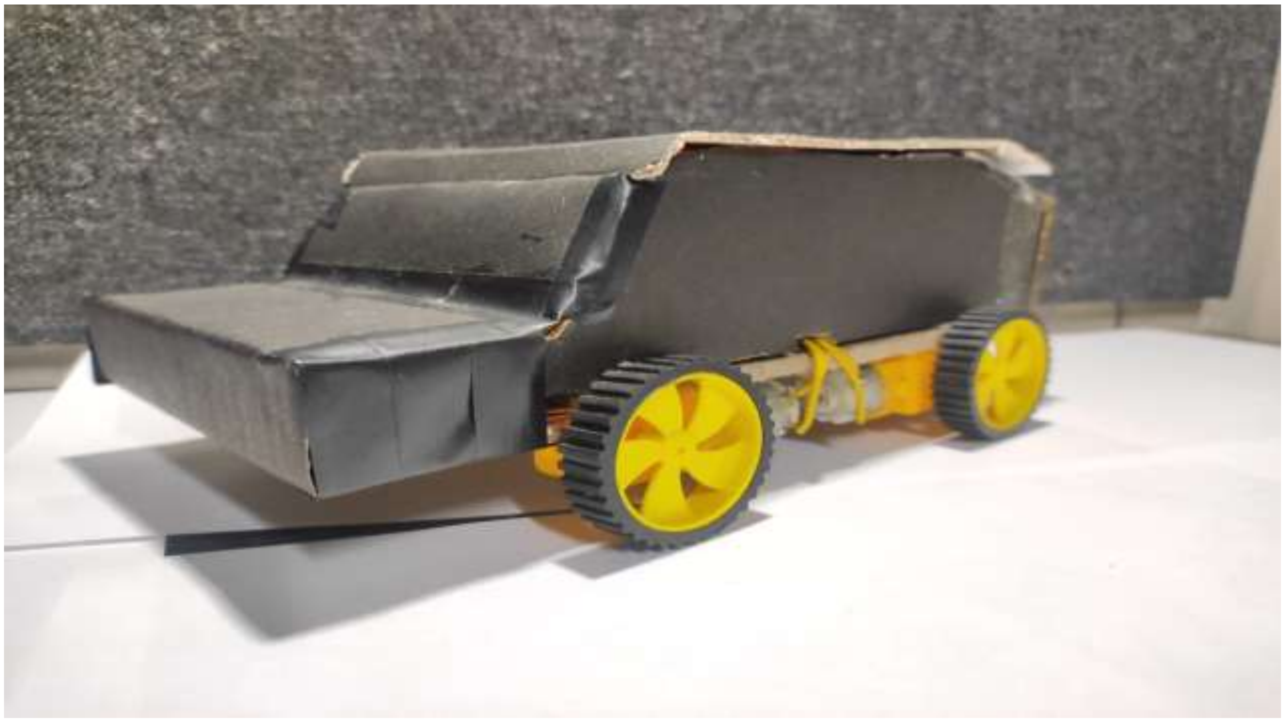

Fig12: MOBILE CONTROLLED ROBOT CAR

Control the Robot Car through the G sensor of our mobile phone and you will be able to move the Robot just by tilting the Phone. We will also use Arduino and RemoteXY app for this G-Sensor Controlled Robot.

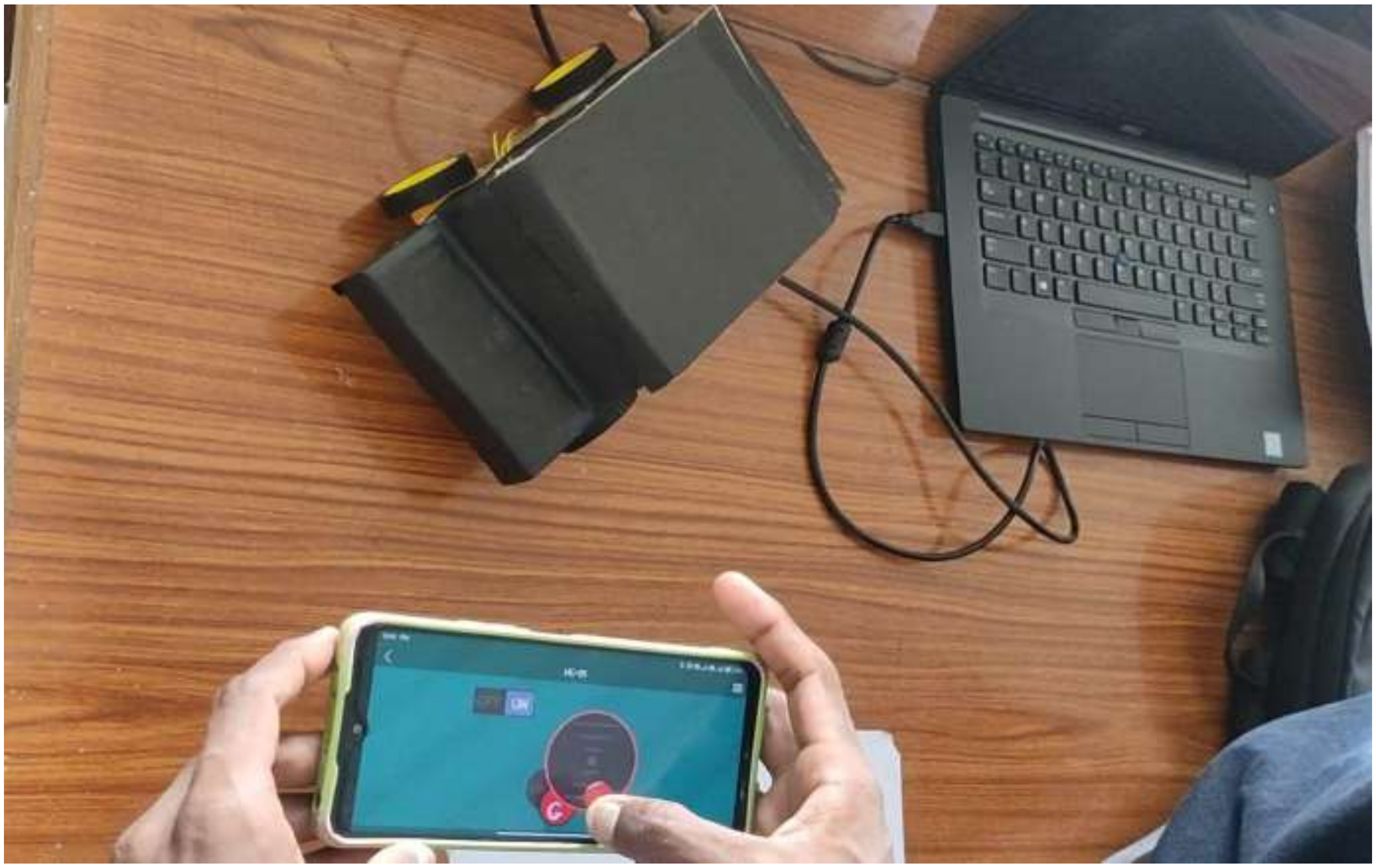

Fig13: MOBILE CONTROLLED ROBOT CAR USING G SENSOR

RemoteXY app is used to create the interface in the Smart Phone for controlling the Robot. We will add the joystick in the interface so that Robot can also be controlled by Joystick as well as by tilting the phone.

CHAPTER 5
CONCLUSION

In this project, the field of Robotics is investigated and a robotic car equipped with many functionalities is proposed. The designed model detects any obstacle encountered in front of it and places it aside. Along with this, explosion sensing is achieved by using a combination of sensors such as Gas sensor, Fire sensor and Temperature sensors. During the project, the complete working of Arduino UNO and a variety of sensors is executed and understood. Coding and designing skills form the base and are utilized in the project. The project gives profound knowledge into different advancements and devices for improvement of the venture. Exposure to various software such as Arduino IDE and Proteus widened the knowledge in a broad aspect. There is much scope of improvement in the communication range and considerable reduction in processing time. The use of Wi-Fi technology instead of the more widely used Bluetooth increases the range of communication and improves the overall performance of the designed model.

REFERENCES

[1] T. L. Chien, H. Guo, K. L. Su and S. V. Shiau, "Develop a Multiple Interface Based Fire Fighting Robot," 2007 IEEE International Conference on Mechatronics, Changchun, Jilin, 2007, pp. 1-6, doi: 10.1109/ICMECH.2007.4280040.

[2] S. Mandal, S. K. Saw, S. Maji, V. Das, S. K. Ramakuri and S. Kumar, "Low cost arduino wifi bluetooth integrated path following robotic vehicle with wireless GUI remote control," 2016 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, 2016, pp. 1-5.

[3] H. Rissanen, J. Mahonen, K. Haataja, M. Johansson, J. Mielikainen and P. Toivanen, "Designing and implementing an intelligent Bluetoothenabled robot car," 2009 IFIP International Conference on Wireless and Optical Communications Networks, Cairo, 2009, pp. 1-6.

[4] Zhao Wang, Eng Gee Lim, Weiwei Wang, M. Leach and Ka Lok Man, "Design of an arduino-based smart car," International SoC Design Conference (ISOCC), Jeju, pp. 175-176, 2014

[5] N. Firthous Begum, P. Vignesh. Design and Implementation of Pick and Place Robot with Wireless Charging Application, International Journal of Science and Research (IJSR), ISSN (Online): 2319-7064, 2013.

[6] Zhenjun He, Jiang Zhang, Peng Xu, Jiaheng Qin and Yunkai Zhu, "Mine detecting robot based on wireless communication with multisensor," 2013 IEEE 4th International Conference on Electronics Information and Emergency Communication, Beijing, 2013, pp. 117- 120.

[7] Y. Zhang, B. K. Chen, X. Liu and Y. Sun, "Autonomous Robotic Pickand-Place of Microobjects," IEEE Transactions on Robotics, vol. 26, no. 1, pp. 200-207, 2010.

[8] S. S. Pujari, M. S. Patil and S. S. Ingleshwar, "Remotely controlled autonomous robot using Android application," International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, pp. 588-593, 2017.

[9] D. Singh, P. Zaware and A. Nandgaonkar, "Wi-Fi surveillance bot with real time audio & video streaming through Android mobile," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology, Bangalore, pp. 746-750, 2017.

[10] M. Meghana et al,2020, Hand gesture recognition and voice-controlled robot, Materials Today: Proceedings, 2214-7853.

[11] Bhanu chandu, Kirupa Ganapathy,2020, Voice Controlled Human Assistence Robot, International Conference on Advanced Computing & Communication Systems (ICACCS), 978-1- 7281-5197-7/20.

[12] Ms. M. Ramjan Begum, Mr. S. Chandramouli, Mr. T. Gowtham,2020, Design And Development Of Dual Axis Control Robot For Writing Robot Through Speech Recognition, International Research Journal of Modernization in Engineering Technology and Science, e- ISSN: 2582-5208.

APPENDIX

```
#define REMOTEXY_MODE__SOFTWARESERIAL

#include <SoftwareSerial.h>

#include <RemoteXY.h>


#define REMOTEXY_SERIAL_RX 2

#define REMOTEXY_SERIAL_TX 3

#define REMOTEXY_SERIAL_SPEED 9600


unsigned char RemoteXY_CONF[] =

  { 3,0,23,0,1,5,5,15,41,11

  ,43,43,1,2,0,6,5,27,11,5

  ,79,78,0,79,70,70,0 };


struct {

  signed char joystick_1_x;

  signed char joystick_1_y;

  unsigned char switch_1;

  unsigned char connect_flag;

}

 RemoteXY;
```

```
#define IN1 10

#define IN2 9

#define ENA 12

#define IN3 8

#define IN4 7

#define ENB 11

#define ledpin 13


unsigned char first_motor[3] =

  {IN1, IN2, ENA};


unsigned char second_motor[3] =

  {IN3, IN4, ENB};


void Speed (unsigned char * pointer, int motor_speed)

{

  if (motor_speed>100) motor_speed=100;

  if (motor_speed<-100) motor_speed=-100;

  if (motor_speed>0) {

    digitalWrite(pointer[0], HIGH);

    digitalWrite(pointer[1], LOW);

    analogWrite(pointer[2], motor_speed*2.55);
  }
  else if (motor_speed<0) {
    digitalWrite(pointer[0], LOW);
    digitalWrite(pointer[1], HIGH);
    analogWrite(pointer[2], (-motor_speed)*2.55);
  }
  else {
    digitalWrite(pointer[0], LOW);
    digitalWrite(pointer[1], LOW);
    analogWrite(pointer[2], 0);
  }
}
void setup()
{
  pinMode (IN1, OUTPUT);
  pinMode (IN2, OUTPUT);
  pinMode (IN3, OUTPUT);
  pinMode (IN4, OUTPUT);
  pinMode (ledpin, OUTPUT);
  RemoteXY_Init ();
```

```
}

void loop()
{
  RemoteXY_Handler ();
  digitalWrite (ledpin, (RemoteXY.switch_1==0)?LOW:HIGH);
  Speed (first_motor, RemoteXY.joystick_1_y - RemoteXY.joystick_1_x);
  Speed (second_motor, RemoteXY.joystick_1_y + RemoteXY.joystick_1_x);
```