CHAPTER 1

PROJECT OVERVIEW

1.1 Introduction

Group consists of four member – Prashant kumar , Santosh kumar , Raj bardwaj , Vishant raj. Our group is worked to create Metal Detector Robotic Vehicle(MDRV) which will be capable of driving on land in order to find small metallic object and our internal project guide **Dr. Fahim Ansari**, Professor of Electrical Engineering.

1.2 Project Specification

This design project will be an amphibious, remote-controlled vehicle. The vehicle, named MDRV, will be equipped with a number of devices a metal detector , Bluetooth , 2 channel relay for the purpose of finding metal objects . MDRV is designed to be useful for defence army and hobbyists who might need to find items lost .

The aim of this project is to control the operation of movement of a remote controlled robot to detect metals.

The main objectives of this project are to use finding metal ahead while moving control by mobile application through Bluetooth.

- The movement of robot is controlled by the transmission of signals through air.
- The receiver senses these signals from the air.
- This mini project makes use of the transmitter and receiver at that is available at low cost hence making it very complicated.
- The Bluetooth based control proves to be more advantageous and the operating range to only a few meters of distance.

CHAPTER 2

MICROCONTROLLER

2.1 General Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

2.2 Features of ATMEL 89C51 Microcontroller

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters

- Six Interrupt Sources
- Programmable Serial Channel and Low-power Idle and Power-down Modes

2.3 Block Diagram ATMEL 89C51 Microcontroller

Fig 2.1 Arduino uno

CHAPTER 3

Relay

3.1 Two channel Relay Module -

A relay is defined as an electrically operated switch; their main use is controlling circuits by a low-power signal or when several circuits must be controlled by one signal. The first relay was used in long distance telegraph circuits as amplifiers, basically they repeated the signal they received from one circuit, and transmitted it into a different one, they were also used in early computers to perform logical operations.

The Arduino relay module is designed for a wide range for micro controllers such as the Arduino board, AVR, PIC, ARM, with digital outputs. This module incorporates 2 relays. The following forms the relay system:

- **Input:** Vcc, connected to the 5V current on the Arduino Board, GND, connected to the ground and 2 digital inputs. (In1 & In2
- **Output**: The 2 channel relay module could be considered like a series switches: 2 normally Open (NO), 2 normally closed (NC) and 2 common Pins (COM).

**NC- Normally Closed, in which case NC is connected with COM when INT1 is set low and disconnected when INT 1 is high.*

**NO- Normally Open, in which case NO is disconnected with COM when INT1 is set low and connected when INT 1 is high.*

Fig 3.1 2-channel Relay Module

3.2 How does it work?

Through the Arduino Software, we can easily code when the relay receives power, it works just as an LED:

Relay Simple Use

digitalWrite (Relay, HIGH); // Relay receives power for 1s

delay (1000);

digitalWrite (Relay, LOW); // Relay doesn't receive power for 1s

delay (1000);

As mentioned earlier, the relay is formed by a series of switches, when receiving power, the NC switch would open and the NC would close. By attaching two LEDs at every switch, we would be able to see a visual representation of what is happening.

3.3 Features:

The 2-Channel Relay Module includes the following features:

Number of Relays	2
Control Signal	TTL Level
Rated Load	7A/240VAC 10A/125VAC 10A/28VDC
Contact Action Time	10ms/5ms
Interface Board	5V 2-Channel Relay interface board, and each one need 15-20mA Driver Current.
Equipment	· Equipped with high-current relay, AC250V 10A; DC30V 10A · Indication LED's for Relay output status.
Supported Microcontrollers	Standard interface that can be controlled directly by microcontroller (Arduino, 8051, AVR, PIC, DSP, ARM, ARM, MSP430, TTL logic)

Fig 3.2 2-Channel Relay Module features

CHAPTER 4

REQUIREMENTS FOR BLUETOOTH COMMUNICATION

4.1 Bluetooth Module

Fig 4.1 HC-05 Bluetooth Module

Our Group Using 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 Rluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

Bluetooth communication is required for the transmission of radio waves from mobile transmitter (remote) to receiver (robot) to enable the movement of the robot in this project. The basic requirements for the bluetooth communication used in this project are as follows:

- Power supply

- Bluetooth mpdule Receiver(HC-05)
- Microcontroller

4.2 Bluetooth Module pins

The data is received by the Bluetooth Module from the antenna pin and this data is available on the data pins. Two Data pins are provided in the receiver module. Thus, this data can be used for further applications.

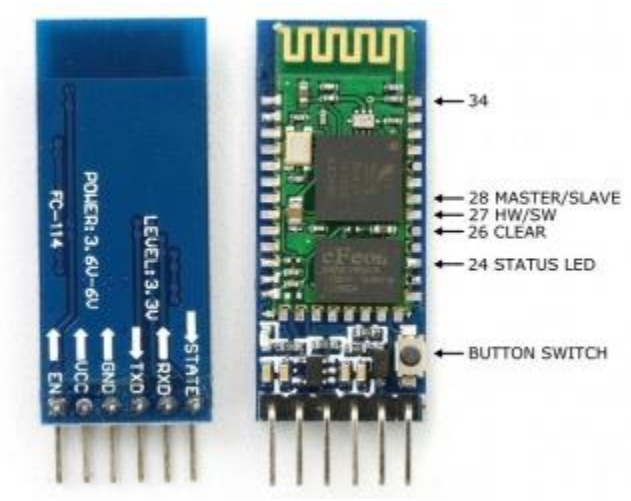

Figure 4.2 Pin diagram of Bluetooth Module

Pin Description

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

- **Vcc**: Supply Voltage 3.3V to 5V
- **GND**: Ground pin
- **TXD & RXD**: These two pins acts as an UART interface for communication
- **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.

- **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.

CHAPTER 5

METAL DETECTOR CIRCUIT

5.1 Circuit Description of metal detector using arduino:

The simplest form of a metal detector consists of an oscillator producing an alternating current that passes through a coil producing an alternating magnetic field. If a piece of electrically conductive metal is close to the coil, eddy currents will be induced in the metal, and this produces an alternating magnetic field of its own. If another coil is used to measure the magnetic field (acting as a magnetometer), the change in the magnetic field due to the metallic object can be detected.

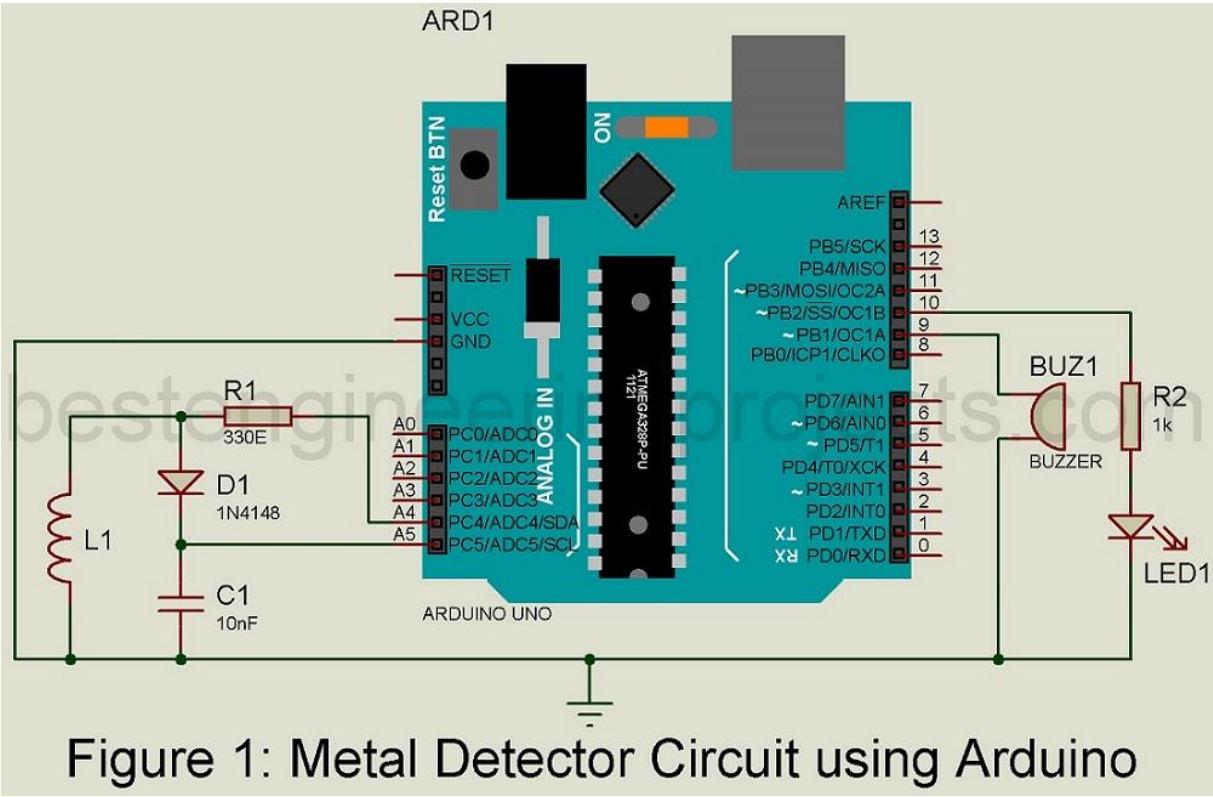

Fig 5.1 Metal Detector Circuit using Arduino

CHAPTER 6

SOFTWARE REQUIREMENT

6.1 Arduino ide

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Fig 6.1 Arduino ide

CHAPTER 7

PROGRAMMING (CODE)

7.1 code of metal Detector

```
#include<SoftwareSerial.h>
#define vinPin A5
#define buz 9
#define pulsePin A4

#define led 10

long sumExpect=0; //running sum of 64 sums
long ignor=0;   //number of ignored sums
long diff=0;        //difference between sum and avgsum
long pTime=0;
long buzPeriod=0;



#define IN1 4
#define IN2 5
#define IN3 6
#define IN4 7
//#define EN1 6
//#define EN2 5

SoftwareSerial mySerial(0, 1); // RX, TX
```

```
String data;
int btVal;

void setup()
{
  Serial.begin(9600);
  pinMode(pulsePin, OUTPUT);
  digitalWrite(pulsePin, LOW);
  pinMode(vinPin, INPUT);
  pinMode(buz, OUTPUT);
  digitalWrite(buz, LOW);
  pinMode(led, OUTPUT);

  //Serial.begin(115200);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  //pinMode(EN1, OUTPUT);
  //pinMode(EN2, OUTPUT);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, HIGH);
  //analogWrite(EN1,63);
  //analogWrite(EN2,63);
  mySerial.begin(9600);
}

void loop()
```

```
{
  int minval=1023;
  int maxval=0;
  long unsigned int sum=0;
  for (int i=0; i<256; i++)
  {
    //reset the capacitor
    pinMode(vinPin,OUTPUT);
    digitalWrite(vinPin,LOW);
    delayMicroseconds(20);
    pinMode(vinPin,INPUT);
    applyPulses();

    //read the charge of capacitor
    int val = analogRead(vinPin); //takes 13x8=104 microseconds
    minval = min(val,minval);
    maxval = max(val,maxval);
    sum+=val;

    long unsigned int cTime=millis();
    char buzState=0;
    if (cTime<pTime+10)
    {
      if (diff>0)
        buzState=1;
      else if(diff<0)
        buzState=2;
    }
    if (cTime>pTime+buzPeriod)
    {
      if (diff>0)
```

```
   buzState=1;
   else if (diff<0)
   buzState=2;
   pTime=cTime;
  }
  if (buzPeriod>300)
  buzState=0;

  if (buzState==0)
  {
   digitalWrite(led, LOW);
   noTone(buz);
  }
  else if (buzState==1)
  {
   tone(buz,2000);
   digitalWrite(led, HIGH);
  }

  else if (buzState==2)
  {
   tone(buz,500);
   digitalWrite(led, HIGH);
  }
}

//subtract minimum and maximum value to remove spikes
sum-=minval;
sum-=maxval;

if (sumExpect==0)
```

```
  sumExpect=sum<<6; //set sumExpect to expected value
  long int avgsum=(sumExpect+32)>>6;
  diff=sum-avgsum;
  if (abs(diff)<avgsum>>10)
  {
    sumExpect=sumExpect+sum-avgsum;
    ignor=0;
  }
  else
    ignor++;
  if (ignor>64)
  {
    sumExpect=sum<<6;
    ignor=0;
  }
  if (diff==0)
    buzPeriod=1000000;
  else
  buzPeriod=avgsum/(2*abs(diff));


while (mySerial.available())
 {
  {
      data = mySerial.readStringUntil('\n');
      //Serial.print(str);
  }

    btVal = (data.toInt());
    //Serial.print("BlueTooth Value ");
    //Serial.println(btVal);
```

```
switch (btVal)
 {
    case 1:
      //Serial.println("Forward");
      forward();
      break;

    case 2:
     //Serial.println("Reverse");
      reverse();
      break;

    case 3:
     //Serial.println("Left");
      left();
      break;

    case 4:
      //Serial.println("Right");
      right();
      break;

    case 5:
      //Serial.println("Stop");
      stoprobot();
      break;

}
```

```
 }


   if (mySerial.available() < 0)
    {
     //Serial.println("No Bluetooth Data ");
    }
}

void applyPulses()
{
    for (int i=0;i<3;i++)
    {
      digitalWrite(pulsePin,HIGH); //take 3.5 uS
      delayMicroseconds(3);
      digitalWrite(pulsePin,LOW);  //take 3.5 uS
      delayMicroseconds(3);
    }
}


void forward()
{
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void reverse()
```

```
{
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void left()
{
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void right()
{
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, HIGH);
}

void stoprobot()
{
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, HIGH);
}
```

CHAPTER 8

Block Diagram and Hardware Requirements

8.1 Block Diagram

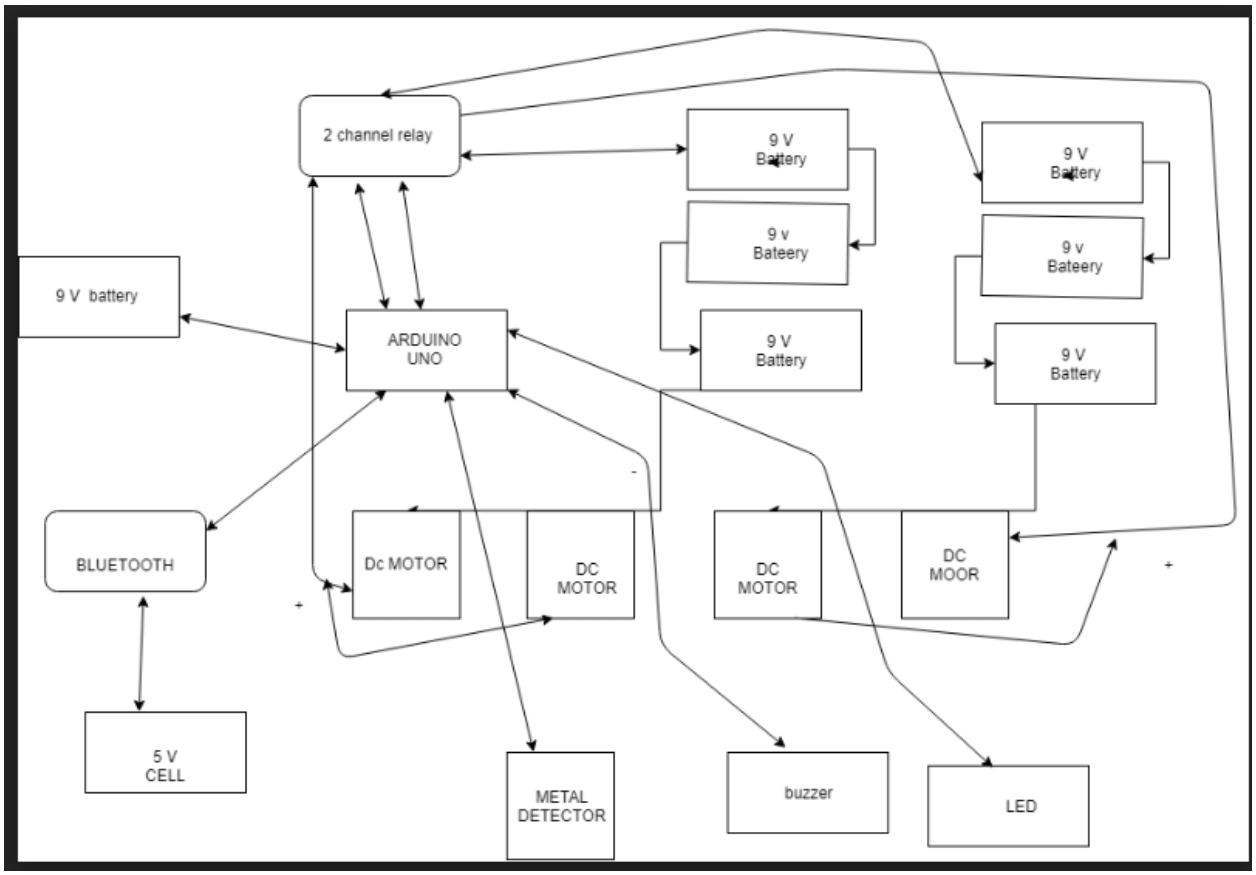

Fig 8.1 Block Diagram

8.2 Hardware Requirements

Arduino Uno , 2 channel Relay , 9V batteries , 4 motors , cells , Bluetooth , metal detector , buzzer and LED.

8.3 Software Requirement

Arduino ide

8.4. DC Motor

Fig 8.2 DC Motor

Here we are going to use 2 dc motor. The dc motor is a low cost and high quality motor. In order to ensure longer life and to reduce wear and tear, we use brass gears and steel pinions. DC motors have been commonly used in many industrial applications such as steel rolling mills, electric cranes, electric vehicles and robotic manipulator due to wide, precise, simple and continuous control characteristics. In many applications motors should be precisely synchronized to give the desired performance.

8.5 Power Supply System

We using 9 V batteries for motor , arduino uno , bluettoth .

8.6 Microcontroller

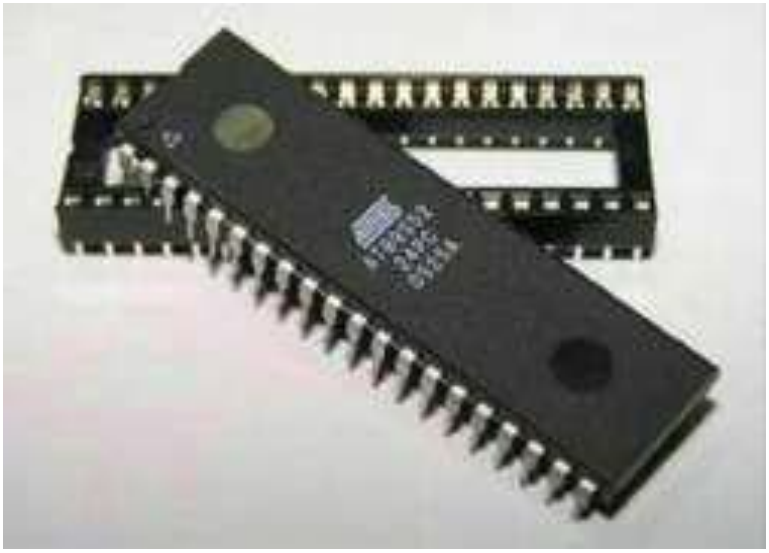

Fig 8.3 Microcontroller

In this project we are using AT89S52 microcontroller chip. A microcontroller is a small computer on a single integrated circuit. In modern terminology it is a system on a chip. Microcontroller is a self-contained system with peripherals, memory and a processor that can be used in a embedded system.

Features

- Working voltage: 4.5V ~ 5.5V
- 12 clocks per machine cycle
- General 8052 family compatible.
- 32K byte on chip program flash
- 1024 byte on-chip data RAM
- Three 16 bit Timers/Counters
- Four 8-bit I/O ports

- A serial I/O port
- One Watch Dog Timer
- Full duplex serial channel

8.7. Two channel Relay Module

A relay is defined as an electrically operated switch; their main use is controlling circuits by a low-power signal or when several circuits must be controlled by one signal. The first relay was used in long distance telegraph circuits as amplifiers, basically they repeated the signal they received from one circuit, and transmitted it into a different one, they were also used in early computers to perform logical operations.

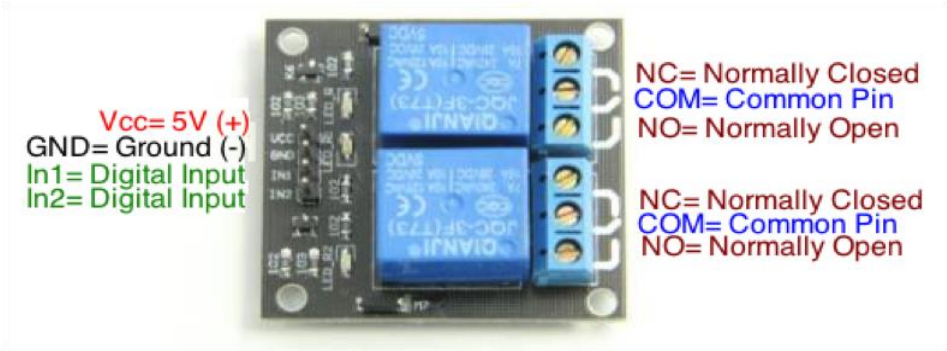

Fig 8.4 2-Channel Relay Module

8.8 Metal Detector Circuit

The simplest form of a metal detector consists of an oscillator producing an alternating current that passes through a coil producing an alternating magnetic field. If a piece of electrically conductive metal is close to the coil, eddy currents will be induced in the metal, and this produces an alternating magnetic field of its own. If another coil is used to measure the magnetic field (acting as a magnetometer), the change in the magnetic field due to the metallic object can be detected.

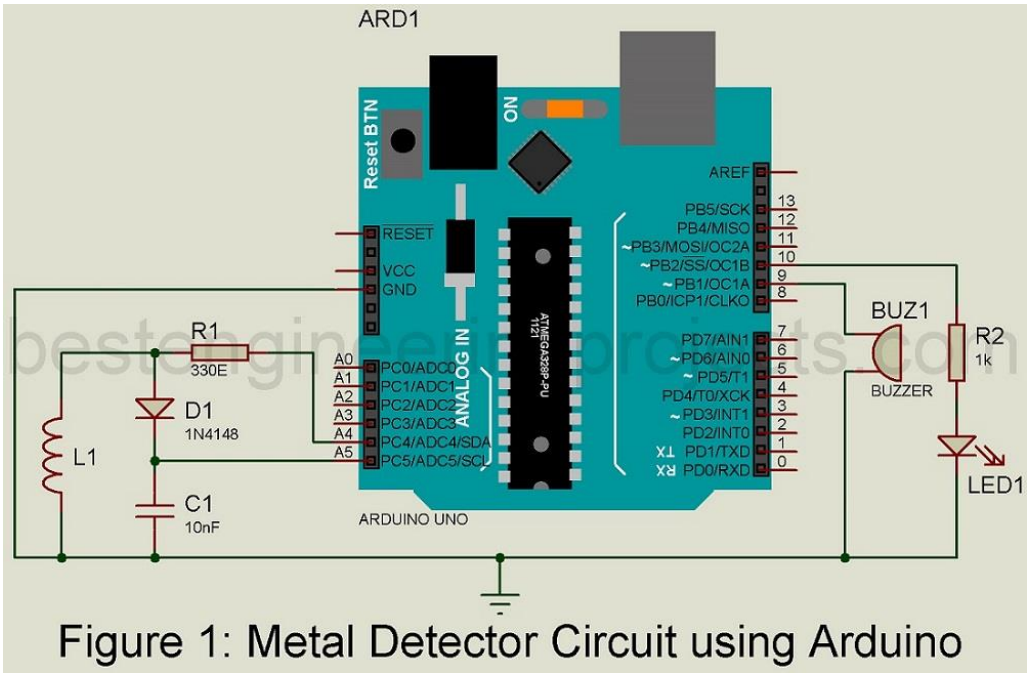

Fig 8.5 Metal Detector Circuit using Arduino

8.9 Arduino ide

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

Fig 8.6 Arduino ide

8.10 Buzzer and LED

Piezo buzzer is an electronic device commonly used to produce sound. Light weight, simple construction and low price make it usable in various applications like car/truck reversing indicator, computers, call bells etc. Piezo buzzer is based on the inverse principle of piezo electricity discovered in 1880 by Jacques and Pierre Curie. It is the phenomena of generating electricity when mechanical pressure is applied to certain materials and the vice versa is also true. Such materials are called piezo electric materials. Piezo electric materials are either naturally available or manmade. Piezoceramic is class of manmade material, which poses piezo electric effect and is widely used to make disc, the heart of piezo buzzer. When subjected to an alternating electric field they stretch or compress, in accordance with the frequency of the signal thereby producing sound.

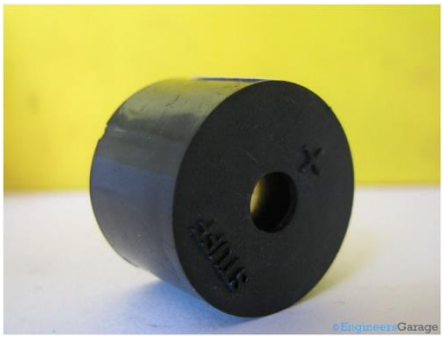

Fig 8.7 Buzzer

A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it.

Fig 8.8 LED

CHAPTER 9

WORKING OF THE PROJECT

This project (Remote Controlled Metal Detecting Robot) consists of two sections-

mobile (Remote) and Receiver section (Robot).

9.1 Mobile (Remote)

In the transmitter section (remote), we have the following components:

- Two switches(motor driver shield)
- Bluetooth module

There are three switches for the movement of the robot in various directions like forward, left and right. These four switches(relay) then connected to Bluetooth module

Depending on that is been pressed (left, right, forward), the digital data from the mobile is transferred to the Bluetooth module, which encodes this digital data to the receiver (robot) through the antenna.

9.2 Receiver (Robot)

The receiver section consists of the following components:

- Bluetooth module
- Microcontroller (AT89C51)
- 2 channel relay module

- Geared motors
- Metal detecting circuit

In this receiver section, Bluetooth module is connected to the microcontroller, which is, in turn, connected to the Relay. This relay is connected to the geared motors of the robot. Metal detecting circuit is also other separate sub-section on the receiver part.

When the radio waves are transmitted from the mobile (remote) to the receiver (robot), these waves are received by the Bluetooth module through the receiver antenna. From the **Bluetooth** receiver the signals are sent to microcontroller, which depending on the code written in it, enables either disable. This Relay correspondingly activates the specified geared motors (Four Geared Motors-). Power set –reset button is used for activating the receiver section. Metal detecting circuit is other sub-section on the receiver part.

As the robot moves in any specified direction and if a metal is been placed on the path of the robot., the inducting coil present at the lower side of the receiver section, which acts as a metal detecting coil, will detect the metal and activates the buzzer sound on the receiver section. The metal detection processes goes on by induction of eddy currents in the metal due to the variation in the magnetic fields of the two components-coils and metal.

CHAPTER 10

OPERATING PROCESS

10.1 OPERATING PROCESS

Working process divided into steps for better understanding

STEP 1 –

- Switch ON the MDRV
- Wait 5 seconds to start

STEP 2 –

- Open your mobile download app Bluetooth Controller
- And install it in your device

STEP 3 –

Now setup app these setting

- First click on set key button
- And type these setting into your app shown below
- Click ok button

your setup done now.

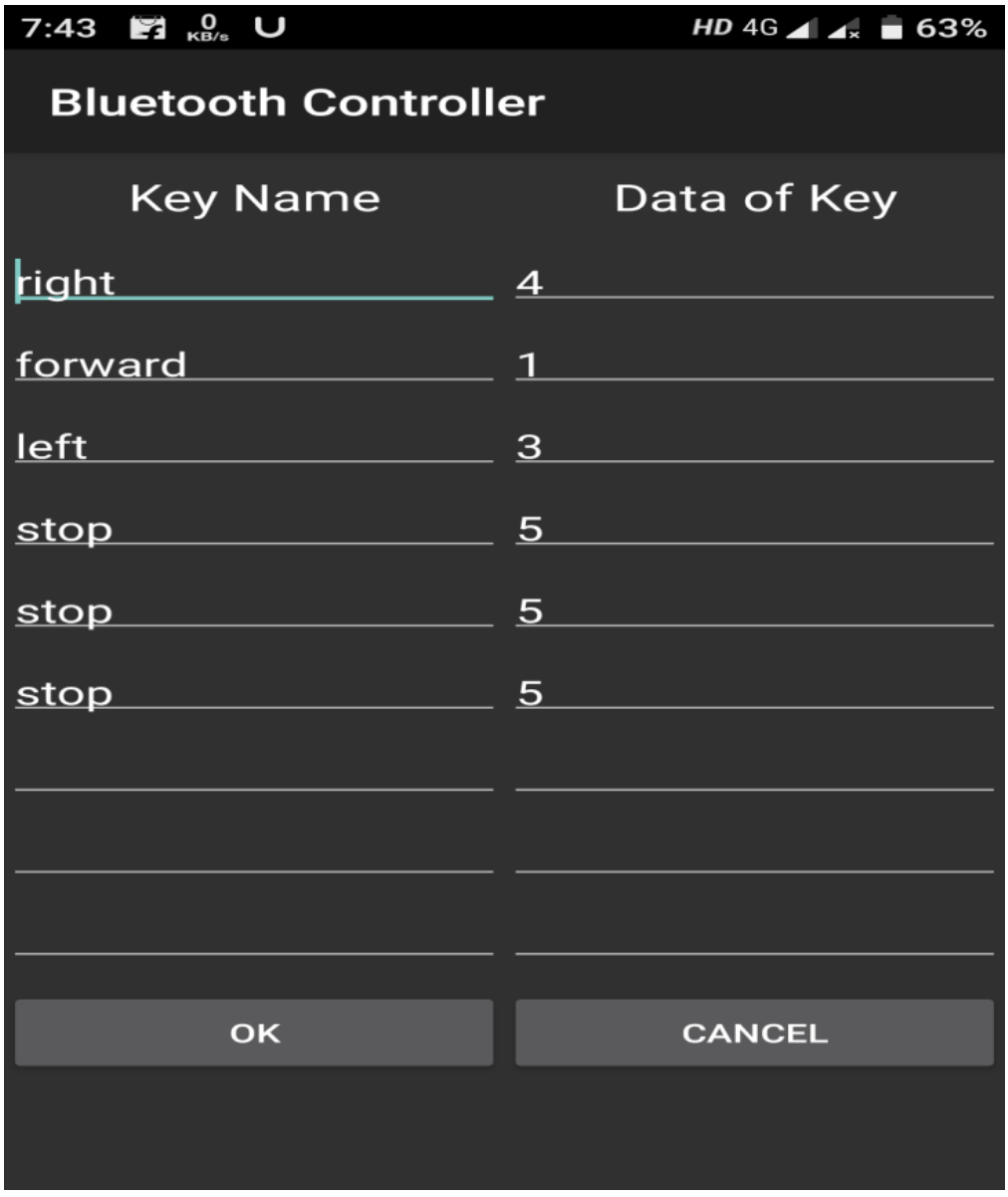

Fig 10.1 Setup the app

STEP 4 –

Now your app look like in below picture right now

Fig 10.2 Setup the app part 2

- click on scan button
- you see their HC-05 click on it

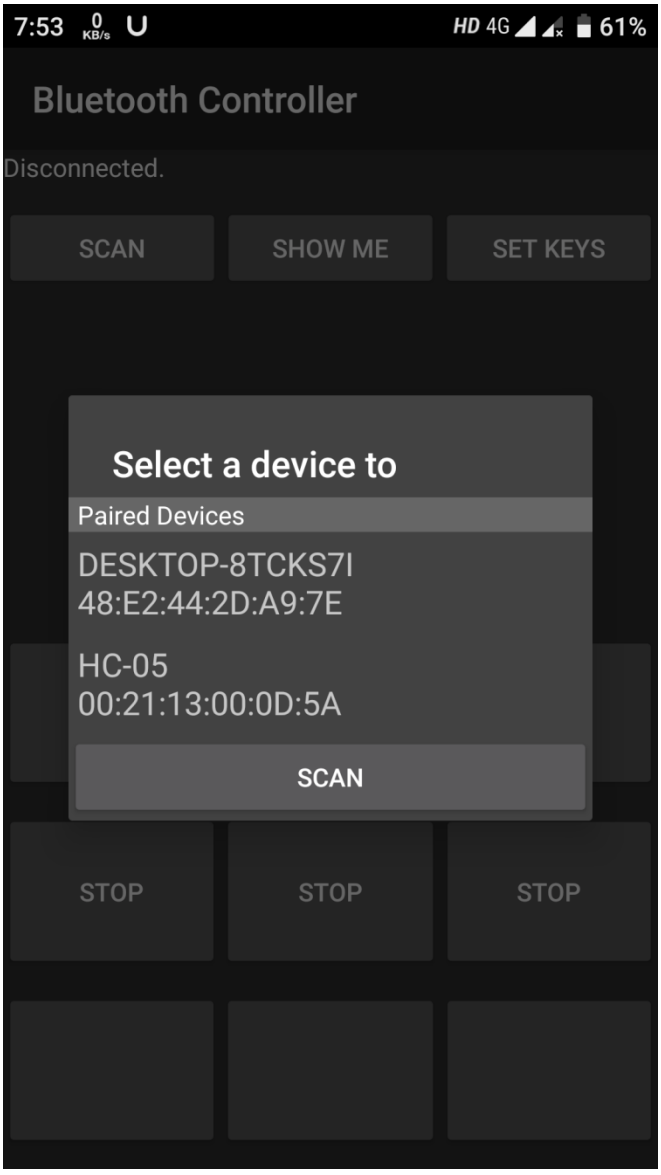

Fig 10.3 Setup the app part 3

- it connect to metal Detector

  NOTE – password(1234)

STEP 5 –

Click on forward button to move forward

Click on right button to move right

Click on left button to move left

Click on stop button to move stop

STEP 6 –

While moving the MDRV it detect metal at the same time and send signal to buzzer

to alert us that there is metal.

STEP 7 –

To turn OFF the MDRV pressed again Switch buthon.

CHAPTER 11

RESULT AND FUTURE SCOPE

RESULT

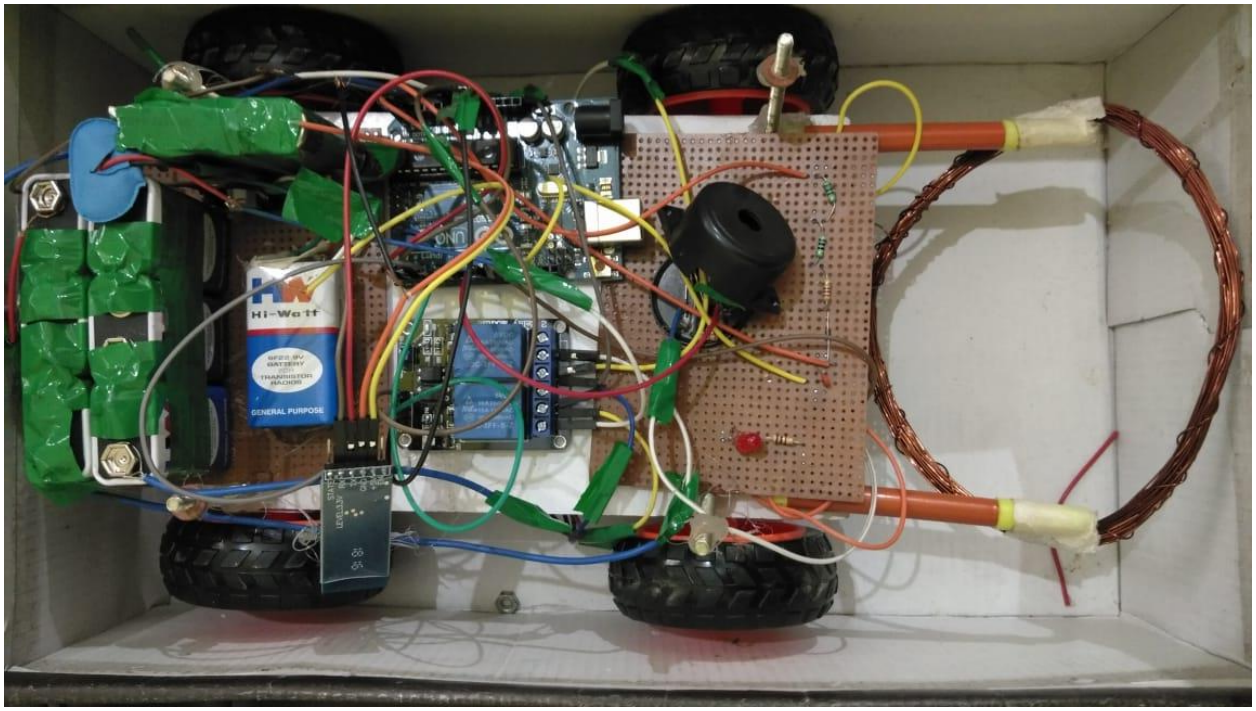

Fig 11.1 Whole Setup of MDRV Front Side

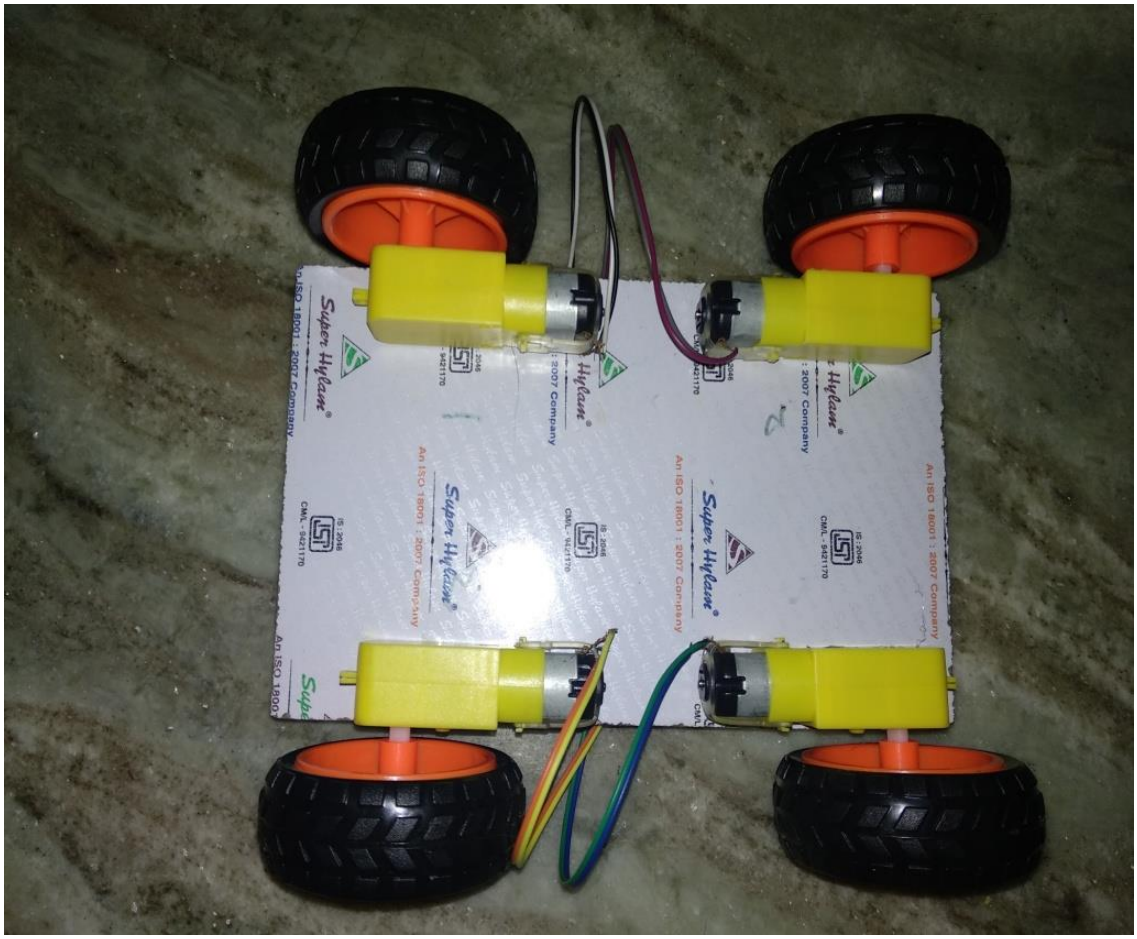

Fig 11.2 Whole Setup of MDRV Back Side

APPLICATIONS

Mines detection-

This remote controlled metal detecting robot with image transmission can be used for detection of mines in remote and others places to know the location of the mine. Since whenever this robot passes through a mine, it detects the mine and produces a buzzer sound .

CONCLUSION

This project presents a metal detecting robot using bluetooth communication and it is designed and implemented with Atmel 89C51 MCU in embedded system domain.

The mine sensor worked at a constant speed without any problem despite its extension, meeting the specification required for the mine detection sensor.

REFERENCES

Text Books:

[1] Raj Kamal, “Embedded Systems”, Pearson Education Publications, 2007.

[2] Mazzidi, “8051 Microcontroller and Embedded Systems”, Prentice Hall Publications, 2nd Edition, 2005.

[3] Edwin S.Grosvenor and Morgan Wesson,”Alexander Graham Bell: The Life and Times of the Man Who Invented the Telephone “, New York, Abrams, 1997.

Magazines:

[1] Electronics for you

[2] Electrikindia

Web portal:

[1] www.howstuffworks.com.

[2] http://www.atmel.com/dyn/resources/prod_documents/doc0265.pdf.

[3] www.intechopen.com/download/pdf/pdfs_id/68.pdf.