LANDMINE DETECTOR

A mini project report submitted

in partial fulfilment of requirement for the completion of

Smart System Design course

By

V.KEDHARESHWAR RAO	**2005A41149**
G. PRUDHVI RAJ	**2005A42015**
A. AKSHITHA	**2003A51157**

Under the guidance(s) of faculty in-charges

Dr. Laxman Raju Thoutam, Assistant Professor

Dr. Leo Joseph, Associate Professor

Department of ECE

SR University, Ananthasagar, Hasanparthy
Warangal Urban 506371, Telangana, India

INDEX

INTRODUCTION

One person is killed by a landmine every 15 minutes. 26,000 people a year become landmine victims. 5,554 mine casualties took place in 2019. 80% of them are civilians – children representing 43% of the civilian casualties.

Still 61 countries and areas around the world are contaminated by landmines and thousands of people continue living with a daily threat of losing their life or limb.

Demining or **mine clearance** is the process of removing landmines from an area. In military operations, the object is to rapidly clear a path through a minefield, and this is often done with devices such as mine plows and blast waves. By contrast, the goal of *humanitarian demining* is to remove all of the landmines to a given depth and make the land safe for human use. Specially trained dogs are also used to narrow down the search and verify that an area is cleared. Mechanical devices such as flails and excavators are sometimes used to clear mines.

A great variety of methods for detecting landmines have been studied. These include electromagnetic methods, one of which (Ground Penetrating Radar) has been employed in tandem with metal detectors. Acoustic methods can sense the cavity created by my casings. Sensors have been developed to detect vapor leaking from landmines. Animals such as rats and mongooses can safely move over a minefield and detect mines, and animals can also be used to screen air samples over potential minefields. Bees, plants and bacteria are also potentially useful. Explosives in landmines can also be detected directly using nuclear quadrupole resonance and neutron probes.

Detection and removal of landmines is a dangerous activity, and personal protective equipment does not protect against all types of landmine.

Once found, mines are generally defused or blown up with more explosives, but it is possible to destroy them with certain chemicals or extreme heat without making them explode.

DISCUSSION ON PROBLEM

Risk is higher for military or defusing squad personnel during terrorist activities that are administered by landmines. So, we should increase using of remotely guided machines which detects the landmines and conveys the location of mine to us which prevents Human loss.

There is also Risk is for animals and extinction also occurs due to booby traps and landmines for catching them. To deal with these problems we have designed a prototype to detect a landmine without involving the human presence to avoid loss of human and animal lives

DISCUSSION ON SOLUTION

To deal with these problems we have designed a prototype to detect a landmine without involving the human presence to avoid loss of human and animal lives.

We have made a landmine detector using Arduino, copper coil, buzzer. Here ,we are making a sensor that identifies the metal which is near to it.

When the device gets close to the metal it detects the metal close to it because of the electromagnetic waves are transmitted from the copper loop.

With these electromagnetic waves, the circuit gives signal to the Arduino. Arduino detects that it is a metal or not and sends the guidance to the signal and the LED by the code. Then Arduino also sends the information about address of landmine by sharing location from GPS helping to point and diffuse landmine easily.

When a landmine is detected it turns on the buzzer so that we can spot the location of the machine and landmine.

PROJECT DESCRIPTION

Arduino

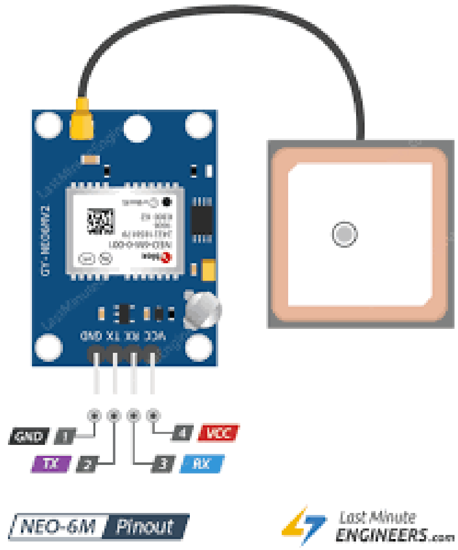
There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phi gets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment,

so students learning to program in that environment will be familiar with how the Arduino IDE works.

- Open source and extensible software - The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries.

NEO-6M GPS

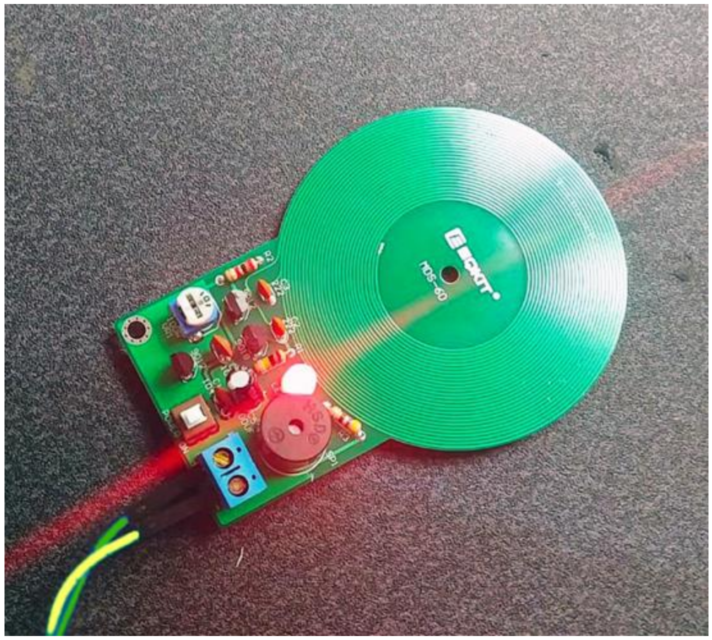


The NEO-6M GPS module is a well-performing complete GPS receiver with a built-in 25 x 25 x 4mm ceramic antenna, which provides a strong satellite search capability. With the power and signal indicators, you can monitor the status of the module. Thanks to the data backup battery, the module can save the data when the main power is shut down accidentally. Its 3mm mounting holes can ensure easy assembly on your aircraft, which thus can fly steadily at a fixed position, return to Home automatically, and automatic waypoint flying, etc

METAL DETECTOR

A **metal detector** is an electronic instrument that detects the presence of metal nearby. Metal detectors are useful for finding metal inclusions hidden within objects, or metal objects buried underground. They often consist of a handheld unit with a sensor probe which can be swept over the ground or other objects. If the sensor comes near a piece of metal this is indicated by a changing tone in earphones, or a needle moving on an indicator. Usually the device gives some indication of distance; the closer the metal is, the higher the tone in the earphone or the higher the needle goes. Another common type are stationary "walk through" metal detectors used at access points in prisons, court houses, and airports to detect concealed metal weapons on a person's body.

The tuned circuit (a coil plus a capacitor across it) formed here generates a smooth sine wave signal. Keep note, the signal produced here comes from the natural ability of the coil and the capacitor across it, and hardly needs any other components. But Q1 helps the tuned circuit to generate the sine wave signal as Q1 supplies the essential pulse of energy at the right moment in each cycle (it turns on at the start of each cycle and delivers the pulse of energy and then turns off).



How does this setup keep running? It's a bit of a complex process that calls for a lengthy explanation

Here, L1 will deliver a signal that increases the noise produced by Q1 to create an oscillator that has a certain amplitude. We can control the amplitude by the trimpot VR1. This signal is passed to Q2 where it gets an uplift and prevents C4 being charged via R2, thus disables Q3 and the piezo-sounder SP1 does not make a noise in idle state.

On the other hand, if a metal object is placed in the close vicinity of the coil (sensor-head), amplitude of the magnetic waves from L2 is reduced somewhat together with the amplitude of the oscillator. This effectively disables Q2 but puts a small voltage across the base and emitter of Q3 to fire it slightly. The piezo-sounder then makes a noticeable tone because a small dc voltage is available across its terminals.

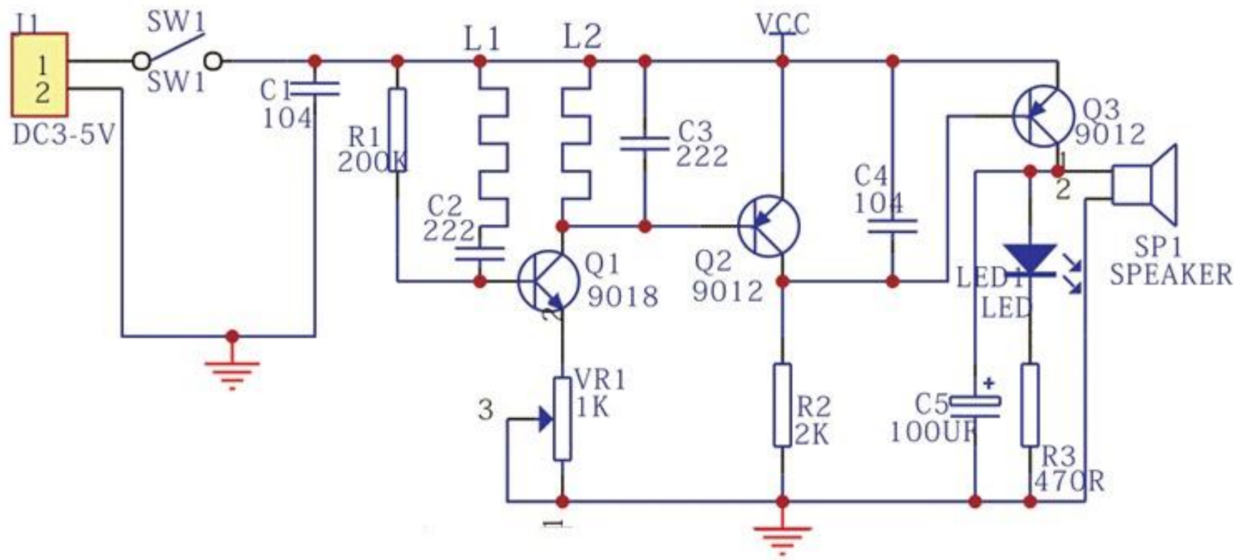
The major advantage of this type of metal detector circuitry is simple construction of both the device and its search-head (coil), but one drawback is its poor sensitivity (very short detection range). This is an ingenious design idea for learning how simple metal detectors/locators work. That's really all there's to it.

we have designed a prototype to detect a landmine without involving the human presence with all these components to avoid loss of human and animal lives.

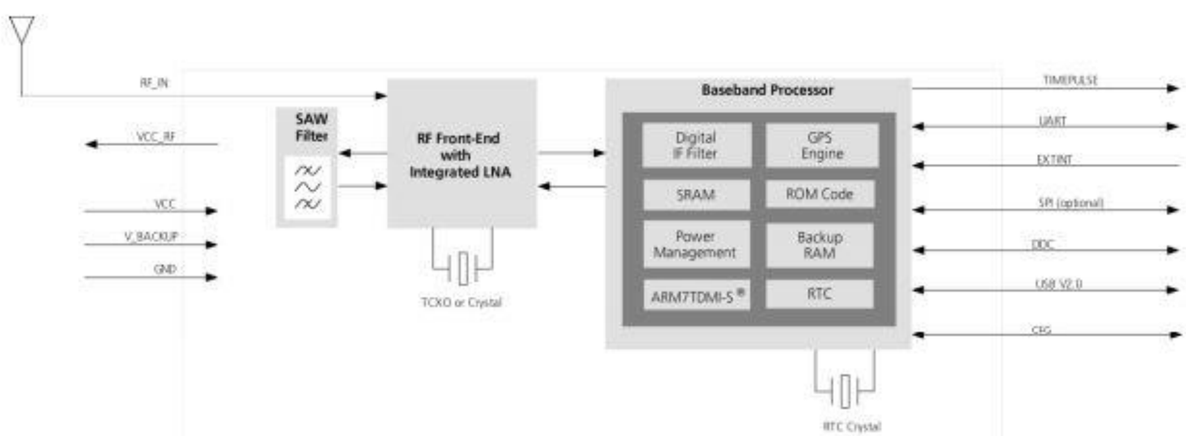
We have made a landmine detector using Arduino, simple metal detector, buzzer. Here, we are making a sensor that identifies the metal which is near to it. When the device gets close to the metal it detects the metal close to it and sends the location to the controller.

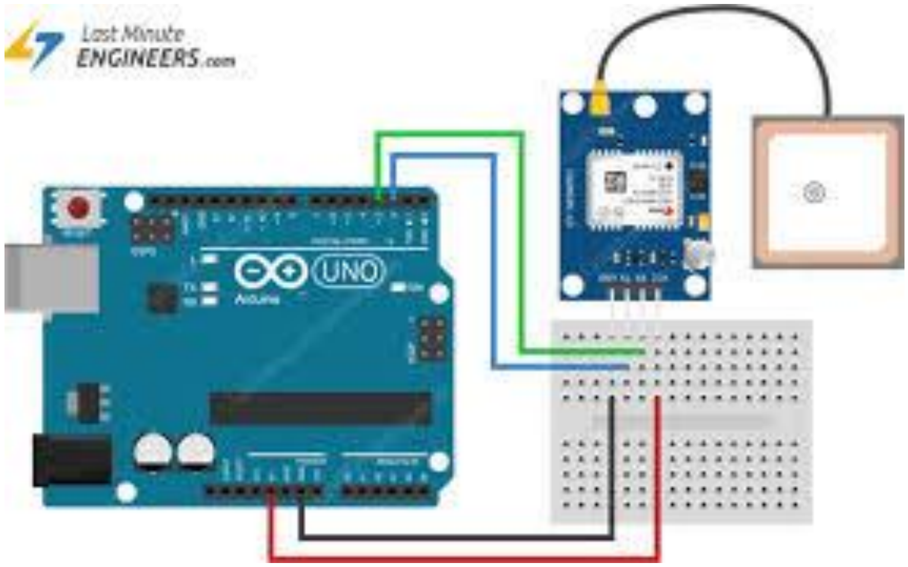
BLOCK DIAGRAMS

BLOCK DIAGRAM OF METAL DETECTOR :



BLOCK DIAGRAM OF GPS MODULE :




Last Minute
ENGINEERS.com
UNO

RESULTS AND DISCUSSION

We can see the output through the serial monitor as of now. we have tried to send the location data through NODEMCU and GSM Module but in vain.

We can also get extra information such as altitude, Exact time, the accuracy of location, satellites connected and more through console applications.

Advantages :

- Circuit diagram is very simple.
- We can easily measure the oscillation resonant frequency of this circuit.
- This circuit can trace an exact location (longitude-latitude) through GPS module ,where the landmine placed?.

Disadvantages :

- It has less sensitivity.
- Even the detectors can’t do anything when they lost the control of Remote car.
- Without signal we can’t work with GPS module.

Conclusion

- This project has been developed considering the need for low cost.
- This landmine detection device can be used in border areas.
- Landmine detection device works effectively.

FUTURE SCOPE

At present, humanitarian demining in most affected areas begins with a United Nations(UN)-led emergency response, which is controlled by ex-pats, who usually have a military background and who are largely paid for by "ear-marked" donations from UN countries. Those donations sometimes take the form of staff and goods. At the same time, as the UN arrives (and sometimes before), the specialist charitably-funded clearance groups, which are funded by an individual government's aid budget or by trusts and donor charities, tend to move into the area.

Mines are placed in different environments (city, jungle, desert) regularly or irregularly and can be redistributed by erosion and surface movements. In 2015 a feasibility study [Gottwald et al, 2015] proved the possibilities of a UAV-based mine detection system for the automatic detection and localization of landmines. Since early 2016, in cooperation with three Swiss and German Universities, a UAV-based system for mine detection has been in development. This system will be used in the process of land release (a very important part of mine action). The system consists simply of a 5kg payload drone, a low-cost RTK-GNSS system, cameras for a photogrammetric production of a DTM, microwave sensors for mine detection and an

anti-collision system.

As technologies are spontaneously updating, a new era of warfare has begun. With help of AI and Machine learning many intelligent systems are being developed every day. If we can use new technologies effectively, we can make fully autonomous rovers or robots which can safely help in the demining process avoiding any human casualities.

REFERENCES

- Youtube links - https://youtu.be/_U1VCQxXSqo

  - https://youtu.be/NFz9tmWeVBA
- Parts - https://thinkrobotics.in

  -https://www.makerkart.com

APPENDICES

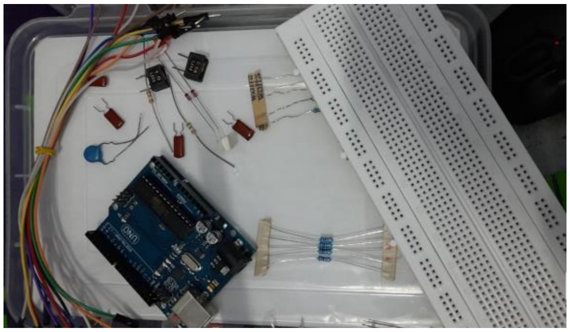


Fig 1: Parts for metal detector

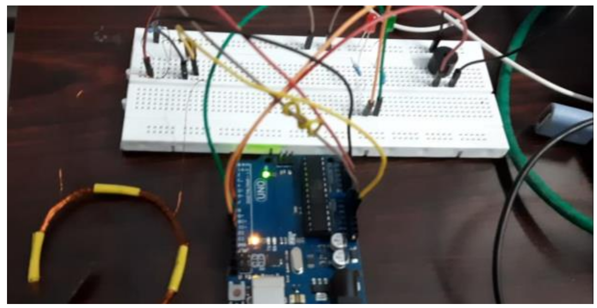


Fig 2: Metal Detector setup

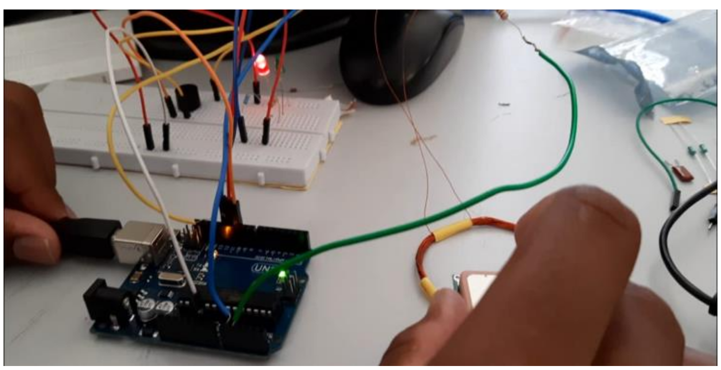


Fig:3 Testing Metal detector

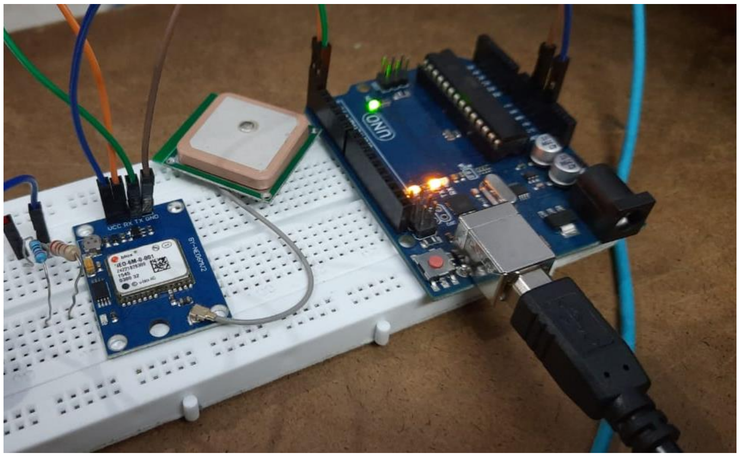


Fig:4 Testing NEO-6M GPS setup

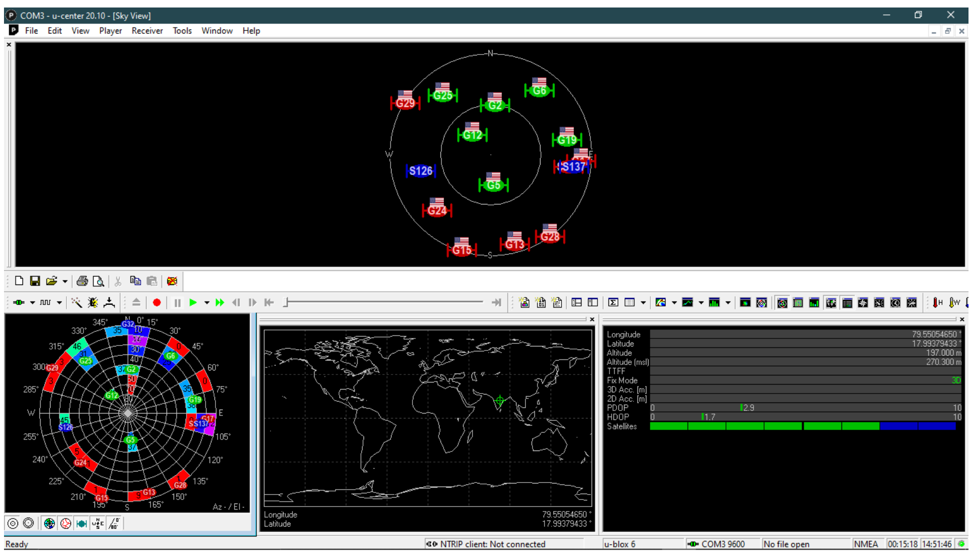


Fig 5: GPS output through Console application

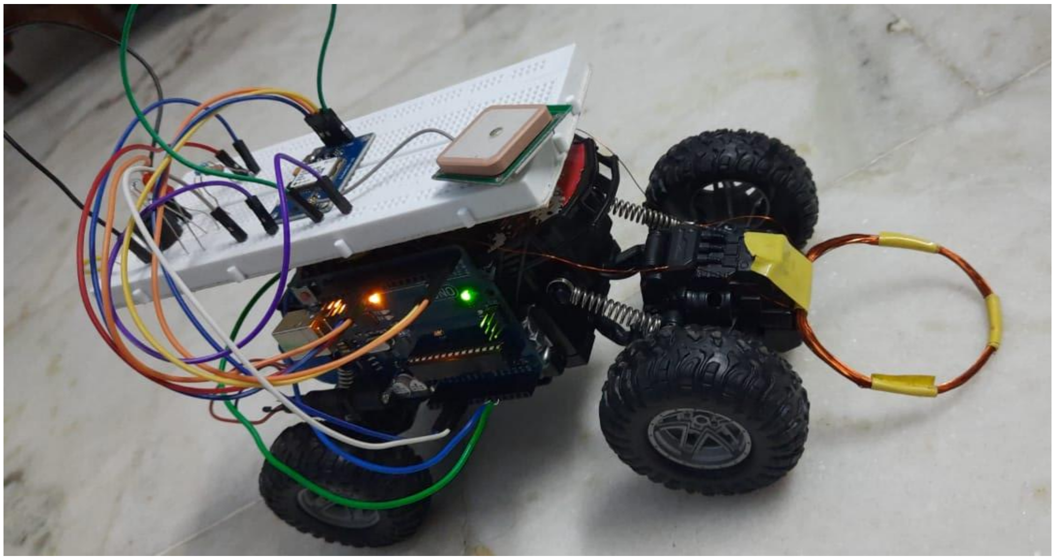


Fig 6: Final Setup

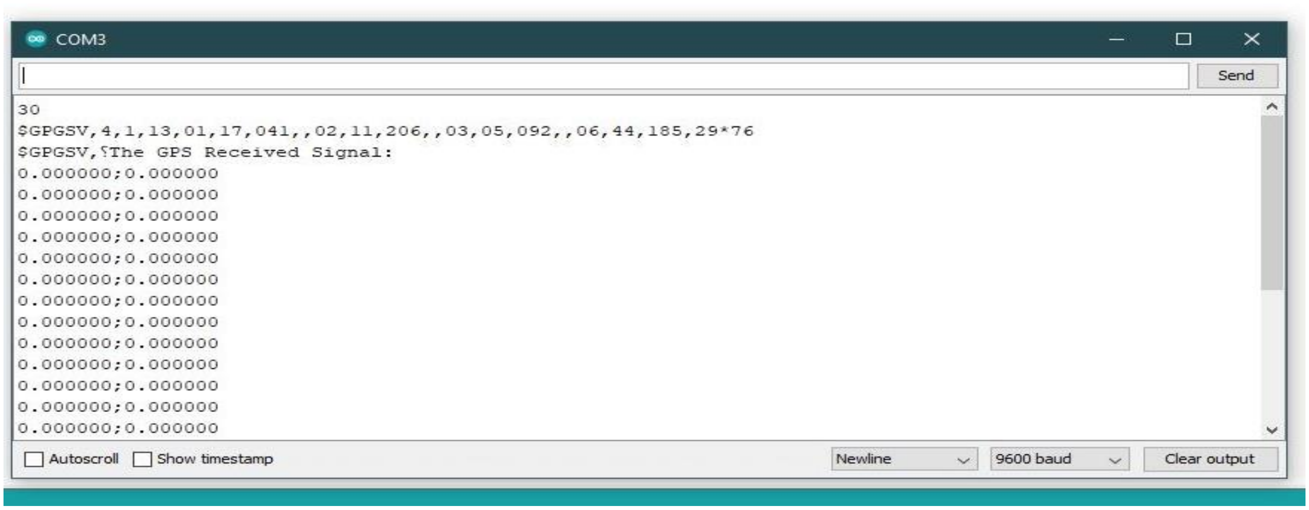


Fig 7: Arduino Serial monitor output

WORKING CODE

```
// wiring:
// 220Ohm resistor on D2
// 10-loop D=10cm seach loop between ground and resistor
// diode (-) on pin A0 and (+) on loop-resistor connection
// 10nF capacitor between A0 and ground
// LED1 in series with 220Ohm resistor on pin 8
// LED2 in series with 220Ohm resistor on pin 9

#include <TinyGPS.h>
#include <SoftwareSerial.h>
const byte npulse = 3;
const bool sound = true;
const bool debug = false;
const byte pin_pulse=A0;
const byte pin_cap  =A1;
const byte pin_LED1 =12;
const byte pin_LED2 =11;
const byte pin_tone =10;
 TinyGPS gps;
SoftwareSerial sw(0, 1); // RX, TX
```

```
float lat ,lon ; // create variable for latitude and longitude object
SoftwareSerial gpsSerial(0,1);//rx,tx

void setup() {
 Serial.begin(9600); // connect serial
 if (debug) Serial.begin(9600);
  pinMode(pin_pulse, OUTPUT);
  digitalWrite(pin_pulse, LOW);
  pinMode(pin_cap, INPUT);
  pinMode(pin_LED1, OUTPUT);
  digitalWrite(pin_LED1, LOW);
  pinMode(pin_LED2, OUTPUT);
  digitalWrite(pin_LED2, LOW);
  if(sound)pinMode(pin_tone, OUTPUT);
  if(sound)digitalWrite(pin_tone, LOW);

Serial.println("The GPS Received Signal:");
gpsSerial.begin(9600); // connect gps sensor

}
const int nmeas=256;  //measurements to take
long int sumsum=0; //running sum of 64 sums
long int skip=0;   //number of skipped sums
```

```
long int diff=0;        //difference between sum and avgsum
long int flash_period=0;//period (in ms)
long unsigned int prev_flash=0; //time stamp of previous flash

void loop() {
 //gps
 while(gpsSerial.available()){ // check for gps data
    if(gps.encode(gpsSerial.read()))// encode gps data
    {
    gps.f_get_position(&lat,&lon); // get latitude and longitude
    // display position
   Serial.print("LAT:");
   Serial.print(lat);
   Serial.println(lat);
   Serial.print(" ");
   Serial.println(lon);
   Serial.print(" ");
   }
  }
  String latitude = String(lat,6);
  String longitude = String(lon,6);
  Serial.println(latitude+";"+longitude);
  delay(1000);
```

```
//gps
//MD
int minval=1023;
int maxval=0;
  //perform measurement
  long unsigned int sum=0;
  for (int imeas=0; imeas<nmeas+2; imeas++){
    //reset the capacitor
    pinMode(pin_cap,OUTPUT);
    digitalWrite(pin_cap,LOW);
    delayMicroseconds(20);
    pinMode(pin_cap,INPUT);
    //apply pulses
    for (int ipulse = 0; ipulse < npulse; ipulse++) {
     digitalWrite(pin_pulse,HIGH); //takes 3.5 microseconds
     delayMicroseconds(3);
     digitalWrite(pin_pulse,LOW);  //takes 3.5 microseconds
     delayMicroseconds(3);
    }
    //read the charge on the capacitor
          int val = analogRead(pin_cap); //takes 13x8=104 microseconds
minval = min(val,minval);
 maxval = max(val,maxval);
```

```
    sum+=val;
    //determine if LEDs should be on or off
  long unsigned int timestamp=millis();
   byte ledstat=0;
    if (timestamp<prev_flash+10){
      if (diff>0)ledstat=1;
      if (diff<0)ledstat=2;
}
    if (timestamp>prev_flash+flash_period){
      if (diff>0)ledstat=1;
      if (diff<0)ledstat=2;
      prev_flash=timestamp;
    }
    if (flash_period>1000)ledstat=0;
    //switch the LEDs to this setting
    if (ledstat==0){
      digitalWrite(pin_LED1,LOW);
      digitalWrite(pin_LED2,LOW);
      if(sound)noTone(pin_tone);
    }
    if (ledstat==1){
      digitalWrite(pin_LED1,HIGH);
   digitalWrite(pin_LED2,LOW);
```

```
      if(sound)tone(pin_tone,2000);
    }
    if (ledstat==2){
      digitalWrite(pin_LED1,LOW);
      digitalWrite(pin_LED2,HIGH);
      if(sound)tone(pin_tone,500);
    }
  }
  //subtract minimum and maximum value to remove spikes
  sum-=minval; sum-=maxval;
  //process
     if (sumsum==0) sumsum=sum<<6; //set sumsum to expected value
  long int avgsum=(sumsum+32)>>6;
  diff=sum-avgsum;
  if (abs(diff)<avgsum>>10){      //adjust for small changes
    sumsum=sumsum+sum-avgsum;
    skip=0;
  }
else {
    skip++;
  }
  if (skip>64){     // break off in case of prolonged skipping
    sumsum=sum<<6;
```

```
    skip=0;
  }
  // one permille change = 2 ticks/s
  if (diff==0) flash_period=1000000;
  else flash_period=avgsum/(2*abs(diff));
  if (debug){
    Serial.print(nmeas);
    Serial.print(" ");
    Serial.print(minval);
    Serial.print(" ");
    Serial.print(maxval);
    Serial.print(" ");
    Serial.print(sum);
    Serial.print(" ");
    Serial.print(avgsum);
   Serial.print(" ");
    Serial.print(diff);
    Serial.print(" ");
    Serial.print(flash_period);
    Serial.println();
  }
}
```