

# Fabrication of a Robot to Detect Railway Track Cracks and Obstacles by Using IOT and GPS

B. Lakshmi<sup>1</sup>, S. Yeswanth<sup>2</sup>, B.Yash pal<sup>3</sup>, T. Sunil<sup>4</sup>

<sup>1</sup>Assistant professor Department of Mechanical Engineering, Adi kavi Nannaya University - University College of Engineering,

<sup>2,3,4</sup> Student Department of Mechanical Engineering, Adi kavi Nannaya University - University College of Engineering,

**Abstract**—The project aims to develop a robust railway crack detection scheme (RRCDS) using ultrasound sensor assembly for railway track geometry surveying. This system uses Wi-Fi and GPS-based technology to detect cracks on railway tracks, reducing accidents and human fatigue. The system sends location of cracks and obstacles via Wi-Fi to mobile phones, allowing for automatic detection without human interference. The advantages include reduced cost, power consumption, and analysis time, promoting better safety standards and effective testing infrastructure.

**Index Terms**—Arduinio, sensors, solar panel, GPS

## I. INTRODUCTION

Railways are crucial for human travel and economic growth in India, but they often face rail accidents due to faults in rail tracks. These accidents can result in the loss of lives, damage to rail infrastructure, and decreased community morale. To prevent these accidents, a robotic vehicle has been proposed for railway applications. The Robust Railway Crack Detection Scheme (RRCDS) uses an Ultra Sound sensor and an infrared sensor to detect cracks on railway tracks. The micro-controller identifies cracks and points their location on the track, reducing resources and time spent on track maintenance. The robot then stops at the area for a brief period before continuing the inspection. The RRCDS is cost-effective, low power consumption, and enhances safety standards for railway tracks. It also features live monitoring via a camera module and GPS tracking. The project aims to revolutionize railway track safety by implementing an automated crack detection system, ensuring safer rail travel.



Fig 1 :3 D Representation of the project

## II. ARDUINO MICRO CONTROLLER

Arduino, founded by Massimo Banzi and David Cuartielles in 2005, is an open-source micro-controller kit used for communication and controlling various devices.



Fig 2: Micro controller ESP8266

**Arduino Architecture:** Arduino uses Harvard architecture with separate memory for program code and data. It has 32 KB of flash memory for code. The Arduino Uno features 14 digital I/O pins, 6 analog inputs, a 16 MHz oscillator, USB connection, power jack, ICSP header, and a reset button.

**Power Jack:** Arduino can be power either from the pc through a USB or through external source like adaptor or a battery. It can operate on a external

supply of 7 to 12V. Power can be applied externally through the pin  $V_{in}$  or by giving voltage reference through the IOREf pin.<sup>[31]</sup>

**Digital Inputs:** It consists of 14 digital inputs/output pins, each of which provide or take up 40mA current. Some of them have special functions like pins 0 and 1, which act as Rx and Tx respectively, for serial communication, pins 2 and 3-which are external interrupts, pins 3,5,6,9,11 which provides pwm output and pin 13 where LED is connected.<sup>[32]</sup>

**Analog input:** It has 6 analog input/output pins, each providing a resolution of 10 bits.

**ULTRASONIC SENSOR:** One of the most popular ultrasonic sensors would be the HC-SR04. The configuration pin of HC-SR04 is VCC (1), TRIG (2), ECHO (3), and GND (4). The supply voltage of VCC is 5V and you attach TRIG and ECHO pin to any Digital I/O in your Arduino Board to power it.<sup>[33]</sup>



Fig 3: Ultrasonic Sensor

### Specifications

- Power Supply: DC 5V
- Working Current: 15mA
- Working Frequency: 40Hz
- Ranging Distance : 2cm – 400cm/4m
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm

### III. IR SENSOR

An infrared sensor detects heat and motion by measuring infrared radiation. It uses an IR LED to emit light and an IR photo-diode to detect it. Changes in resistance and output voltage indicate the amount of infrared light received.



Fig 4 : Infrared Sensor

**DC MOTOR:** A DC motor is a type of motor that changes direct current into mechanical energy, often using magnetic forces. They are widely used in tools, toys, and appliances, and can have their speed adjusted easily. Larger DC motors power electric vehicles and elevators. AC motors are gradually replacing them in some areas.



Fig 5: DC Motor

**RELAY:** Relays are switching that open and close circuits electronically and electro-mechanically. They control circuit contacts, changing states when energy is supplied. Relays are used in control panels and automation to manage power and small currents. They can also switch large voltages with low voltage at the coil. Preventive relays detect issues like over current and overload to protect equipment, and are used for heating elements, alarms, starting coils, and lights.

**Relay Function:** The text describes the relay diagram to explain relay wiring and circuits. It details how the relay works with an iron core, electromagnet, and control switch. When the coil is energized, it attracts contact arms to close the circuit, while de-energizing creates an open circuit. The movable arm returns to its original position due to gravity and spring force. Relays serve two main purposes: applying high voltage to reduce arcing and low voltage to minimize circuit noise.

### Features

- The coil of a relay passes a relatively large

- current, typically 30mA for a 12V relay,
- It can be as much as 100mA for relays designed to operate from lower voltages.
- The maximum output current for the popular 555 timer IC is 200mA so these devices can supply relay coils directly without amplification.<sup>[46]</sup>

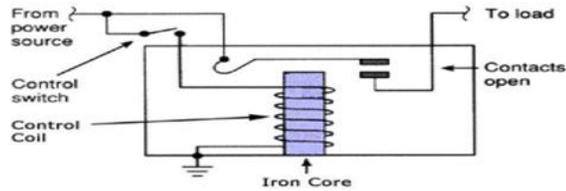


Fig 6: Relay Circuit



Fig 7: Relay chip

Wi-Fi Module (NODE MCU) : The ESP8266 WiFi Module connects micro-controllers to WiFi and can host applications or handle networking. It comes with AT command firmware for easy use with Arduino, offering WiFi similar to a shield. It's affordable and supported by a large community. The module has strong processing and storage capabilities for easy integration with sensors. It requires little external circuitry and supports VoIP and Bluetooth. Community resources are available to help users with the ESP8266, including instructions for IoT projects.

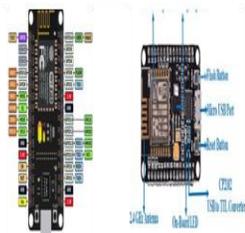


Fig 8a & 8b : Wifi modules

Note: The ESP8266 Module is not capable of 5-3V logic shifting and will require an external Logic Level Converter. Please do not power it directly from

your 5V dev board. This new version of the ESP8266 WiFi Module has increased the flash disk size from 512k to 1MB.<sup>[51]</sup>

Features:

- 802.11 b/g/n
  - Wi-Fi Direct (P2P), soft-AP
  - Integrated TCP/IP protocol stack
  - Integrated TR switch, balun, LNA, power amplifier and matching network
  - Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode.<sup>[52]</sup>

GPS MODULE: GPS stands for "Global Positioning System." It is a satellite navigation system that provides location and time information in all weather. GPS is used for navigation in various vehicles like planes, ships, cars, and trucks. It offers important services to both military and civilian users around the world, delivering real-time, 3D positioning, navigation, and timing globally.



Fig 9: GPS

BATTERY: An electric battery is a power source with one or more electrochemical cells, providing energy for devices. The negative terminal gives electrons, while a redox reaction generates electrical energy. The term "battery" now also covers single-cell devices.



Fig 10: Battery

**SOLAR PANEL:** A solar panel is made of multiple photovoltaic cells in a frame, using sunlight to create direct current electricity. They operate on the photovoltaic effect, where light exposure causes materials to release electrons. Individual cells aren't powerful alone, so they are grouped into modules and arrays to supply electricity.



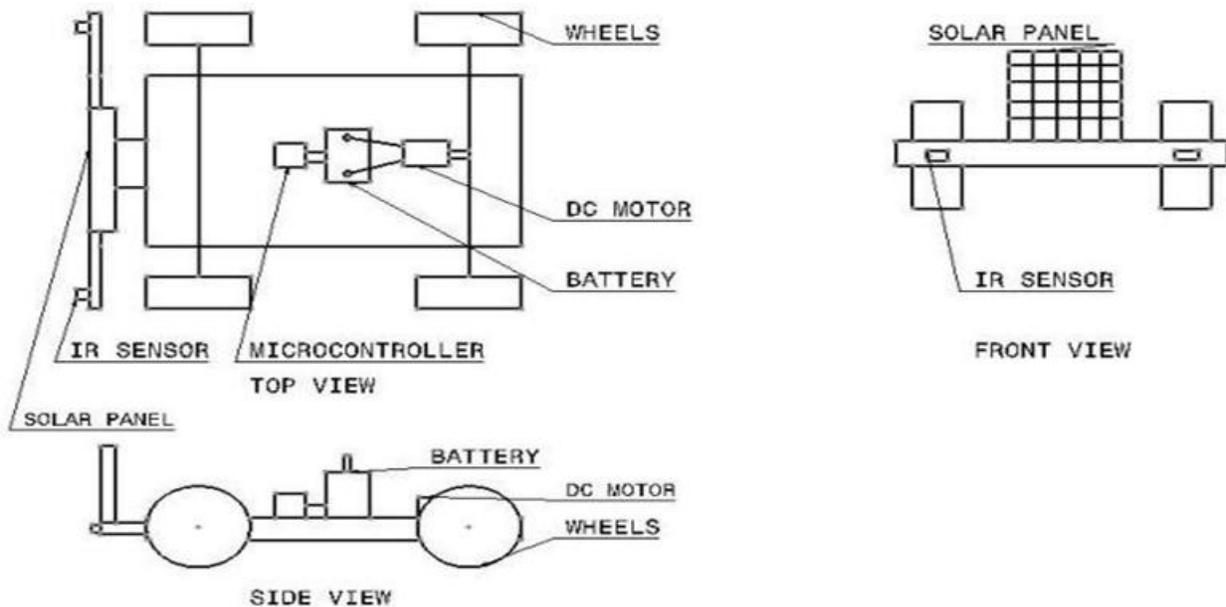
Fig 11: Solar panel

controller for image and video processing projects. It supports various resolutions and works with the ESP-IDF and Arduino IDE. It can capture images, record videos, and perform image processing. The module also has features like face detection and object recognition, making it useful for IoT devices and surveillance systems.



Fig 12: Camera

**ESP32CAMERA MODULE:** The ESP32 camera module combines a camera with the ESP32 micro-



$$L \times B = 3\text{ft} \times 2\text{ft}$$

Fig 13: Construction Model

**STEPS INVOLVED IN MAKING PROTOTYPE :**

Step 1 : Build the Autonomous robot by using IOT and sensors.

Procure the parts as planned to build the robot and assemble the chassis by using cast iron beams

Step 2 : Assemble the parts and connect the arduino  
Create a flat-form with base to place the parts and

connect them by using cables

Step 3 : Code the arduino and assemble the battery with solar panel.

Connect the arduino with IOref pin to a computer and upload the code fix the battery with solar panel

Step 4 : Testing and error decoding

After assembly test the prototype and check for errors to resolve for gaining efficiency

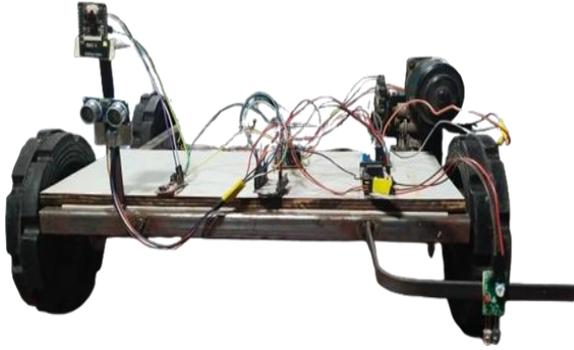
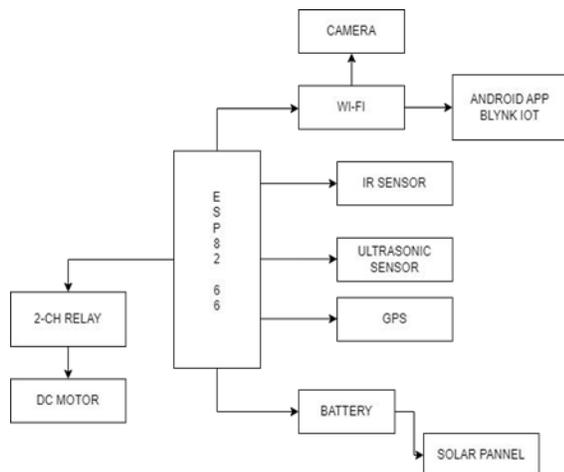


Fig 14: Final output

#### IV. RESULTS AND DISCUSSIONS

Addition of Arduino, sensors and an Esp32 cam to the mechanism can allow for more sophisticated control and automation, while the Ultra sound sensor can detect nearby objects and potentially trigger a response from the mechanism and IR sensor can detect the cracks in the railway track. The combination of an Arduino, Esp32 cam, Ultra sound sensor and IR sensor can create a unique and interactive robotic sculpture with potential applications in railways, engineering and education. The results are to check the working condition of the chip and sensor. The sensor is set to identify the cracks by 50 cm. The sensor is also tuned to identify the obstacles by 50cm. The results of the experimentation are successfully stored in cloud and executed the commands programmed in the chip. The results are also used to identify the signals transmitted by the chip to sensor and from sensor to chip via command language input.



Flow Chart 1 : Ploy of Project

#### V. CONCLUSIONS

Railway crack detection by using ultrasound transmitter and receiver robot is an eminent way to detect the crack in the railway track. The defect information which includes GPS value is wireless transmitted to the railway safety management centre through GSM Module and the alert can be given to the next approaching train.

The proposed scheme possesses many advantages such as fast monitoring and reporting system, low cost, low power consumption and less analysis time. Also the easy availability of the components make an ideal project for industrial use with very little initial investment. So the current location device on rail track can easily be measured from home station

#### REFERENCES

- [1] Assembly,” IEEE Int. Conf. on Networking, Sensing and Control, vol. 6, iss. 3, pg. 453-460, May2012
- [2] Conf. on Networking, Sensing and Control, vol. 14, no. 4, pp. 961-970, April 2008
- [3] K. Vijayakumar, S.R. Wylie, J. D. Cullen, C.C. Wright, A.I. Shammaa, “ Non-invasive rail track detection system using Microwave sensor,”
- [4] Qiao Jian-hua; Li Lin-sheng; Zhang Jing-gang; “Design of Rail Surface Crack- detecting System Based on Linear CCD Sensor,” IEEE Int.
- [5] Ravi Shankar Shekhar, Purushottam Shekhar, Ganesan P, “Automatic Detection of Squats in Railway Track”, IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems. vol. 3, iss. 6, pg. 413-413 December 2015
- [6] Richard J. Greene, John R. Yates and Eann A. Patterson, “Crack detection in rail using infrared methods,” Opt. Eng. 46, 051013, May 2007
- [7] R. T. Lemmon, —The influence of the number of satellites on the accuracy of RTK GPS positions, || Australian Surv., vol. 44, no. 1,pp. 64–70, 1999.
- [8] Selvamraju Somalraju, Vigneshwar Murali, Gourav Saha, Dr.V.Vaidehi, “Robust Railway Crack Detection Scheme (RRCDS) Using LED

and LDR.

- [9] <https://www.mapsofindia.com/my-india/india/why-doso-many-train-accidents-occur-in-india>
- [10] Z. Mao, Y. Zhan, G. Tao, B. Jiang, and X.G. Yan, "Sensor Fault Detection for Rail Vehicle Suspension Systems with Disturbances and Stochastic Noises", IEEE Transactions on Vehicular Technology, Vol. PP, Issue 99, pp. 1-1, doi: 10.1109/TVT.2016.2628054, 2016.
- [11] S. Faghih-Roohi, S. Hajizadeh, A. Núñez, R. Babuska, B. D. Schutter, "Deep Convolutional Neural Networks for Detection of Rail Surface Defects", 2016 International Joint Conference on Neural Networks (IJCNN), pp. 2584-2589, doi: 10.1109/IJCNN.2016.7727522, 2016.
- [12] Raghupatty, Pranab, Dineshkumar, Bharath," Automatic Rail Detection Using Ultrasonic Detector", International Conference on Modern Trends in Signal Processing, 2012.

Arduino Nano Board Program:

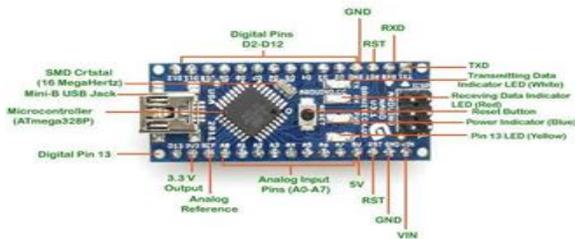


Fig 15: Arduino board

```
#define BLYNK_PRINT Serial #include
<TinyGPSPlus.h> #include <SoftwareSerial.h>
static const int RXPin = D3, TXPin = D4;
static const uint32_t GPSBaud = 9600;

// The TinyGPSPlus object TinyGPSPlus gps;
// The serial connection to the GPS device
SoftwareSerial ss(RXPin, TXPin);
/* Fill-in your Template ID (only if using
Blynk.Cloud) */ #define BLYNK_TEMPLATE_ID
"TMPL3etWYWRsz" #define
BLYNK_TEMPLATE_NAME "project"
#define BLYNK_AUTH_TOKEN
"B6Vv3Hj5DRrlcM0Cf10UX2HeHvfHytV0"
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```

```
// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon). char auth[] =
BLYNK_AUTH_TOKEN;
// Your WiFi credentials.

// Set password to "" for open networks. char ssid[] =
"Abc";
char pass[] = "12345678";

//const int trig_Pin1 = D5;

//const int echo_Pin1 = D6;

//long duration1;

//int distance1;

int obstaclePin = D5; int hasObstacle;
const int trig_Pin2 = D7; const int echo_Pin2 = D8;
long duration2;
int distance2;
int FW = D1; int BW = D2; int count = 0; void setup()
{

// Debug console Serial.begin(9600);
ss.begin(GPSBaud); pinMode(obstaclePin, INPUT);
pinMode(FW, OUTPUT); pinMode(BW, OUTPUT);
digitalWrite(FW, HIGH);
// pinMode(trig_Pin1, OUTPUT); // Sets the trigPin
as an Output

// pinMode(echo_Pin1, INPUT);

pinMode(trig_Pin2, OUTPUT); // Sets the trigPin as
an Output pinMode(echo_Pin2, INPUT);
Blynk.begin(auth, ssid, pass);

// You can also specify server:

//Blynk.begin(auth, ssid, pass, "blynk-cloud.com",
80);

//Blynk.begin(auth, ssid, pass,
IPAddress(192,168,1,100), 8080);

}
```

```

void loop()
{
  Blynk.run();

  while (ss.available() > 0) if (gps.encode(ss.read()))
    displayInfo();
  crack();
  obstacle(); stat();
}

void crack(){
  hasObstacle = digitalRead(obstaclePin);

  // if (hasObstacle == HIGH) {

  // Blynk.virtualWrite(V0,"CRACK DETECTED");

  // digitalWrite(FW,LOW);

  //

  //

  // }

  // else {

  // Blynk.virtualWrite(V0,"NO CRACKS");

  // digitalWrite(FW,HIGH);

  //

  // }

}

//void crack(){

// // Clears the trigPin

// digitalWrite(trig_Pin1, LOW);

// delayMicroseconds(2);

// // Sets the trigPin on HIGH state for 10 micro

seconds

// digitalWrite(trig_Pin1, HIGH);

// delayMicroseconds(10);

// digitalWrite(trig_Pin1, LOW);

// // Reads the echoPin, returns the sound wave
travel time in microseconds

// duration1 = pulseIn(echo_Pin1, HIGH);

// // Calculating the distance
// distance1= duration1*0.034/2;

// // Prints the distance on the Serial Monitor

// //Serial.print("Distance_crack:");

// Serial.println(distance1);

// if (distance1 < 15 ){

//

//

// Serial.println("CRACK DETECTED!");

// digitalWrite(FW, LOW);

// }

// else{

//

//

// Serial.println("NO CRACK!");

// }

//

//}

void stat(){

if(distance2 < 40 || hasObstacle == HIGH){ count = 1;

}

}

```

```

if (count == 1) { digitalWrite(FW, LOW);
}

else{

digitalWrite(FW, HIGH);

}

if (hasObstacle == HIGH) { Blynk.virtualWrite(V0,
"Crack Detected");
}
else {

Blynk.virtualWrite(V0, " ..... ");

}

if (distance2 < 40) {

Blynk.virtualWrite(V3, "Obstacle Detected");

}

else {

Blynk.virtualWrite(V3, " ..... ");

}

}

void obstacle(){

// Clears the trigPin digitalWrite(trig_Pin2, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trig_Pin2, HIGH);
delayMicroseconds(10);
digitalWrite(trig_Pin2, LOW);

// Reads the echoPin, returns the sound wave travel
time in microseconds duration2 = pulseIn(echo_Pin2,
HIGH);
// Calculating the distance distance2=
duration2*0.034/2;
// Prints the distance on the Serial Monitor

//Serial.print("Distance_OBSTACLE:");

Blynk.virtualWrite(V4,distance2);
// if (distance2 < 40 ){

//

// Blynk.virtualWrite(V3,"OBSTACLE
DETECTED!");
// digitalWrite(FW, LOW);

// }

// else{

// digitalWrite(FW, HIGH);

// Blynk.virtualWrite(V3,"NO OBSTACLE");

//

// }

} BLYNK_WRITE(V2) {
if ( param.asInt() ) { count = 0;
}

} BLYNK_WRITE(V1) {
if ( param.asInt() ) { digitalWrite(BW, HIGH);
digitalWrite(FW, LOW);
}

else {

digitalWrite(BW, LOW); digitalWrite(FW, LOW);
}

}

void displayInfo()

{

Blynk.virtualWrite( V10,F("Location: "));

//Serial.println(F("Location: "));
if (gps.location.isValid())

{

```

```
//Serial.print("LATITUDE:");

//Serial.println(gps.location.lat(),           6);
Blynk.virtualWrite(V8,gps.location.lat(),6);
//  Serial.print(F(","));

//  Serial.print("LONGITUDE:");

//      Serial.println(gps.location.lng(),     6);
Blynk.virtualWrite(V9,gps.location.lng(),6);
}

else

{

//Blynk.virtualWrite(V0,F("INVALID"));

//Serial.print(F("INVALID"));
Blynk.virtualWrite(V8,"81.8726");
Blynk.virtualWrite(V9,"17.0656");
}

delay(500);

}
```