E-ISBN: 978-93-7020-967-1

IOT BASED TIME STATUS INDICATING SYSTEM FOR RAILWAY CROSSING GATES

Abstract

Since 20th century there has been an ample growth in the transport system from 20 million to 140 million so does the two-wheeler private transport. The two-wheeler transport had a growth from 14 million to 120 million. There are significant concerns regarding safety and the environment in this situation. Level crossings become overcrowded when a train is delayed and the gates remain closed for a long time. The blockage at the railway level crossing creates a traffic management problem that affects traffic flow performance and traffic safety as well. The inaccurate information about the train running time and the time of crossing the railroad leads to wastage of time because the crossing remains blocked for several minutes. The Railway Crossing Gate Status Detection System is a useful project which detects whether the railway gate is open or closed at all times. This project proposes a method for displaying the gate status on the webpage in the user's android device at that moment, thereby reducing the amount of traffic near the gates and the number of potential accidents that can occur. It completely eliminates two things the first one is saving plenty of time for users and the other one is reducing the traffic jams.

Authors

Vamsi Krishna N
1Senior Assistant Professor,
Electrical and Electronics Engineering,
Velagapudi Ramakrishna Siddhartha
Engineering College, India

Devi Sri Bhanu Prasad K
Student,
Electrical and Electronics Engineering,
Velagapudi Ramakrishna Siddhartha
Engineering College, India
devisribhanuprasadkarapa@gmail.com

Imran SK
Student,
Electrical and Electronics Engineering,
Velagapudi Ramakrishna Siddhartha
Engineering College, India

Nikhil Kumar J
Student,
Electrical and Electronics Engineering,
Velagapudi Ramakrishna Siddhartha
Engineering College, India

Prashanthi B
Student,
Electrical and Electronics Engineering,
Velagapudi Ramakrishna Siddhartha
Engineering College, India

I. INTRODUCTION

Railway crossings are vital points on the railway network where the paths of trains and road vehicles intersect. These crossings serve as crucial junctures that facilitate the safe and efficient movement of both trains and road traffic. However, ensuring the safety of these intersections remains a top priority for railway authorities worldwide. To enhance the safety and operational efficiency of railway crossings, modern technology has played a pivotal role. One such technological advancement is the implementation of a Time Status Indicating System for railway crossing gates. Traditionally, railway crossings have been equipped with manually operated gates or signalmen who lower barriers to block road traffic when a train approaches. While these methods have served their purpose for many years, they are susceptible to human error and may not always provide timely warnings to motorists and pedestrians. Moreover, as traffic volumes increase and train schedules become more complex, the need for a more automated and precise system has become evident. The Time Status Indicating System for railway crossing gates addresses these challenges by incorporating advanced technology, real-time data analysis, and communication systems. This system enhances the safety of railway crossings by providing accurate information about train schedules, gate status to motorists and pedestrians. It not only minimizes the risk of accidents but also contributes to the smooth flow of traffic, reducing congestion and delays at crossings.

II. LITERATURE REVIEW

1. In the plan, ultrasonic sensors and GSM technology are replaced by RF transmitter/receiver pairs to detect approaching trains, IR sensors to detect obstacles, and X Bee transceivers for the railway communication language.

2. The automation of railway gate control using a microcontroller is a remarkable technological innovation that has significantly improved the safety and efficiency of railway crossings. This system relies on a carefully orchestrated set of components. At its core, a microcontroller serves as the central decision-making unit, processing data from a network of sensors including proximity sensors for train detection and traffic sensors for vehicle presence. Actuators, such as motorized gate mechanisms and warning signals, ensure timely gate closure and provide vital alerts to pedestrians and motorists. A communication module facilitates real-time data exchange, and robust control software orchestrates the entire operation. Power supply redundancy and safety features guarantee system reliability and prevent accidents. In essence, this integration of microcontrollers, sensors, actuators, communication modules, software, and safety features represents a groundbreaking advancement in railway safety and automation, promising safer crossings and smoother traffic flow.

3. The system uses 4 infrared sensors to detect trains, LEDs to control traffic, RFID readers to send alerts or warnings, and servo motors to open and close doors. These devices are connected to the Raspberry Pi board as shown in Figure Raspberry Pi controls and operates all these devices according to the program written on Raspberry Pi. When the RFID reader reads the train ID, train information such as date, time, train RFID number is sent to the central server on the Internet. Once data is received from the Pi, the central server instantly retrieves the numbers of all registered trains by RFID number and sends a notification.

4. A switch will be placed at the connection junction of the railway gate with the holder placed on the ground. It acts as a triggering switch for the whole circuit. When the gate is open the switch is open. When the gate is closed the switch is closed, thus completing the circuit. The switch is connected with an Arduino UNO board which is equipped with an ultrasonic sensor(HCSR-04) and a WiFi module(ESP8266) connected with an hotspot to transmit data to the cloud. While the gate is closed the ultrasonic sensor starts to transmit the data about the distance of the object in it's line of sight. Based on the data, a virtual perimeter is set and the ultrasonic sensor monitors for any object near the gate. Once an object/person enters this virtual perimeter, the buzzer starts a alarm to warn the entry. The Arduino decides using the programmer defined conditions to display the status (open or closed) of the gate. This result along with the timestamp will be uploaded in the cloud platform-thingspeak.com, using the wifi-module connected to the Arduino. Retrieving information from cloud, it is posted in an open source website.

5. They use ultrasonic sensors in the system concept. Sensors are installed along the railway. on the receiving end. On the transmitter side, the sensor receives the signal and converts it into electrical energy. Ultrasonic sensors (also called transceivers) work similarly to radar or sonar in that they measure the characteristics of the target by interpreting the echoes of radio waves or sound waves, respectively. Ultrasonic sensors produce sound frequencies and measure the echoes received from the sensor. Motor drivers act like current generators because they use a current control signal and provide a higher current signal. Higher current signal is used to drive the motor There are two sets of H-bridge drivers in the L293D. In operating modes, two DC motors can be driven simultaneously in forward or reverse directions.

6. The main system of our system is to receive training in the use of GPS technology and send the data through the GSM network to a central control center for data processing and processing, review the information and make appropriate decisions. Location information is periodically sent to the central server via the module GSM transmitter. The server updated the information with the correct location, speed and information for each train. The GPS receiver can determine the latitude and longitude and ground speed of a particular train by receiving data from GPS satellites. The device can store data when GSM disconnects and synchronize with the remote server when GSM comes back Online. Using GSM over GPRS effectively increases the performance and availability of the systems. They choose GSM as the communication medium between the train station and the central server to increase the capacity using the existing GSM network, which is the universal service country. Centralized control includes remote control and completes all business data received from train stations via GSM network.

7. The new system they developed has four main components: train detectors, Zig-Bee wireless communication devices, railway crossing gates and monitoring. Detectors are used to detect the arrival and departure of trains. To do this, two infrared sensors are used. A sensor is installed at the station to detect the arrival of the train. When a train is detected, a command to close the door is passed through the train. A detector added to the site will respond to the presence of the object. The sensor will send out a signal and if the signal is not returned, the object does not exist. If the train approaches, the output signal will be sent back and received by the sensor. If there is no problem, it is assumed that there are no trains on the road and the open gate will be sent to the gate to cross the railway.

III. METHODOLOGY

In addressing this challenge, we propose the use of two infrared sensors placed at a specific distance apart. These sensors will calculate the speed of a passing train as it moves between them. Once the speed is determined, we will employ predefined rules to calculate the time taken for the train to cover this distance. Subsequently, we will synchronize this time with the current real-time using an RTC module. The final step involves presenting this time data on a publicly accessible URL through an IoT platform, specifically ThingSpeak, when a gate is closed. This solution offers a systematic approach to measure train transit times accurately, derive train speeds, and display this time information in real time via an IoT platform, thus serving both practical and safety-related purposes.

To accomplish this task, we'll follow a step-by-step approach:

Hardware Setup: Place two infrared sensors along the railway track at a known distance apart. These sensors will detect the presence of a train passing between them. Connect these sensors to a micro controller, such as an Arduino which will be the central processing unit for data collection and calculations. Attach a Real-Time Clock (RTC) module to the micro-controller. The RTC module ensures accurate timekeeping and synchronization with real-world time.

Sensor Data Collection: Write a program for the micro-controller to continuously monitor the two infrared sensors. When a train passes the first sensor, record the timestamp of this event. Speed Calculation: As the train moves between the two sensors, the program measures the time it takes for the train to travel this known distance. Using the formula: Speed = Distance / Time, calculate the speed of the train. With the help of the speed we can able to calculate the time using the ARDUINO code.

Time Calculation: Retrieve the current time from the RTC module, which provides a highly accurate time reference. The time is calculated based upon the speed. Add this relative time to the current time obtained from the RTC module to calculate the absolute time when the train passed between the sensors.

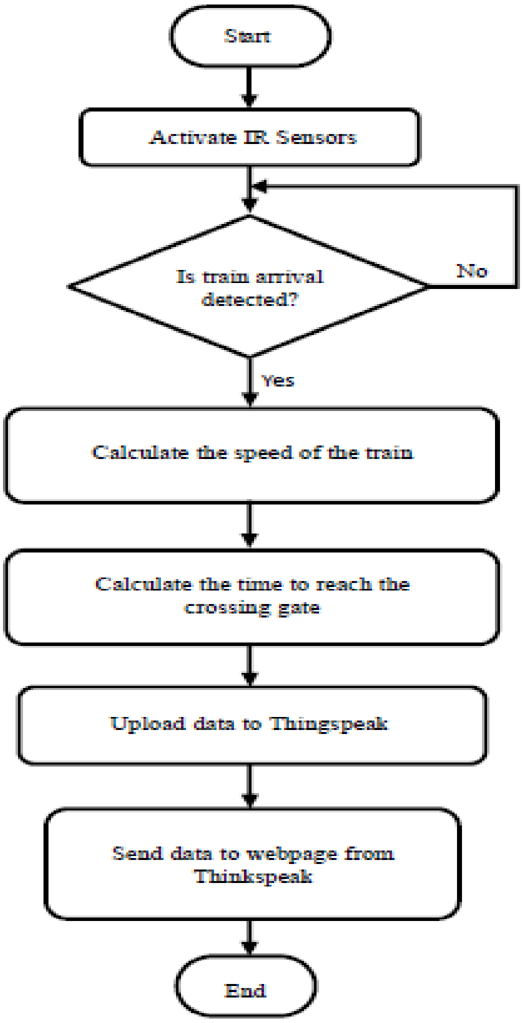

Figure 1: FlowChart Representation of Time Status Indicating System for Railway Crossing Gates

Data Upload: Program the micro-controller to send data to ThingSpeak when certain conditions are met, such as when the gate is closed. Data sent to ThingSpeak should include the calculated time when the train passed between the sensors, the train's speed, and any other relevant information. Data Display: Use ThingSpeak's capabilities to display the collected data on a URL or a web-based dashboard. This URL can be accessed by relevant stakeholders, such as railway operators or the public, to view real-time and historical train passage information.

Safety Measures: Implement safety protocols based on the data collected. For example, you can trigger gate closures or warning signals when a train is detected. By following these steps, you can create a system that accurately measures and records train passage times, calculates train speeds, and displays this information in real-time via an IoT platform. This solution offers both practical applications, such as monitoring train schedules, and safety enhancements by automating gate control based on train movement. A "Time Status Indicating System for Railway Crossing Gates" is a crucial safety application that can help manage railway crossings more efficiently. It involves the use of Arduino boards, along with a NodeMCU (ESP8266), to connect IR (Infrared) sensors and an RTC (Real-Time Clock) module.

IV. COMPONENTS NEEDED

Arduino Board: Arduino is used as a central controller for collecting data from sensors and controlling the gate. NodeMCU (ESP8266): The NodeMCU, which is based on the ESP8266 micro controller, is used for IoT connectivity. It allows the system to communicate with an Online platform (e.g. ThingSpeak) to log data and provide remote access.

IR Sensors: Two IR sensors are placed on either side of the railway track at a certain distance apart. These sensors detect the presence of a train as it passes by. RTC Module: An RTC module provides accurate real-time data, which is crucial for time stamping train arrivals and departures.

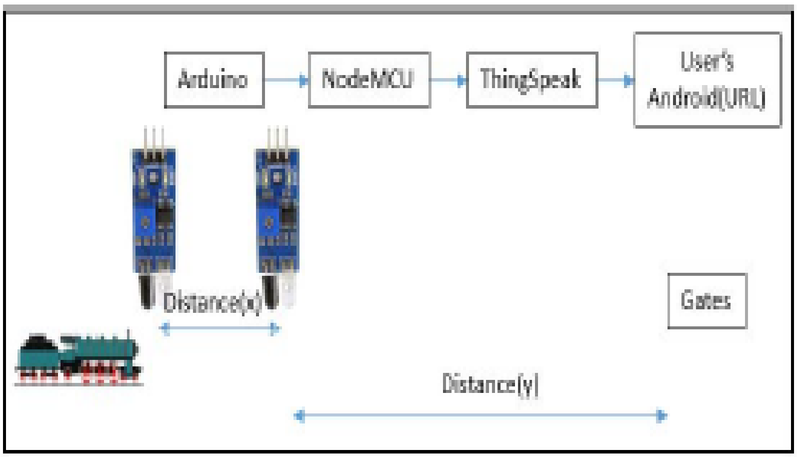

Figure 2: Block Diagram Representation of Time Status Indicating System for Railway Crossing Gates

V. SYSTEM IMPLEMENTATION

1. **Sensor Setup: Install** the two IR sensors on either side of the railway track, facing each other. These sensors are positioned in such a way that when a train passes, it interrupts the IR beam, indicating the presence of a train.

2. **Arduino Board:** The Arduino board is connected to the IR sensors and the RTC module. Write a program for the Arduino to continuously monitor the IR sensors. When a train interrupts the IR beam at the first sensor, the Arduino records the timestamp as the arrival time. When the train interrupts the IR beam at the second sensor, the Arduino records the timestamp as the departure time. NodeMCU (ESP8266): The NodeMCU is connected to the Arduino and serves as the communication bridge between the local system and the IoT platform (e.g., ThingSpeak). The NodeMCU can communicate with the Arduino over a serial connection to receive the train arrival and departure timestamps.

Figure 3: HardWare Setup

3. **Transmission:** The NodeMCU sends this data, including the timestamps and gate status (open/closed), to an IoT platform like ThingSpeak. The IoT platform can be set up to log and display this data on a web page or dashboard accessible via a URL.

4. **Remote Access**: Stakeholders, such as railway operators or the public, can access the real-time status of the railway crossing gate through the URL provided by the IoT platform.

5. **Transfer of Data from ThingSpeak to the URL:** To transfer the status data of the railway gate from ThingSpeak to a URL using HTML and bring the webpage Online using GitHub, you can create a simple HTML webpage that includes python to fetch data from ThingSpeak and then display it on the webpage. This Python Script code can use Fetch API to make an HTTP request to retrieve the status data from ThingSpeak. Once you have created the HTML page, you can upload it to a GitHub repository. By enabling GitHub Pages for that repository, you can make the webpage accessible Online through a URL provided by GitHub. This way, users can visit the URL and view the real-time status of the railway gate on the webpage hosted on GitHub Pages, ensuring easy accessibility and sharing of the information.

VI. RESULTS

The Time Status Indicating System for railway crossing gates utilizes ThingSpeak as a data visualization and IoT platform. It operates by integrating sensors or controllers at railway crossings to gather real-time information about the gate closure schedule. This data is then transmitted to ThingSpeak, which serves as the central hub for displaying this information. Through the ThingSpeak interface, users, including both railway personnel and commuters, can access up-to-date information regarding the scheduled closing time of the railway crossing gates. This system enhances safety and efficiency by offering a digital and easily accessible means of conveying critical information, ensuring that individuals can plan their journeys while railway authorities can effectively manage and coordinate gate closures. The Time Status Indicating System for railway crossing gates, with its display of scheduled gate closure times through a GitHub-hosted URL, represents a modern and technology driven solution to enhance safety and efficiency in railway operations. This system likely integrates IoT sensors or controllers at railway crossings, which continuously collect data regarding gate closure schedules. This data is then securely stored within a GitHub repository, benefiting from the platform's version control and data management features. The GitHub-hosted URL serves as the bridge between the collected data and the end-users, including both railway staff and the general public. This web based interface provides an intuitive and easily accessible means for individuals to access real-time information about when the crossing gates are expected to close.

Figure 4: The time at which the gate is scheduled to be closed is displayed in ThingSpeak(IoT)

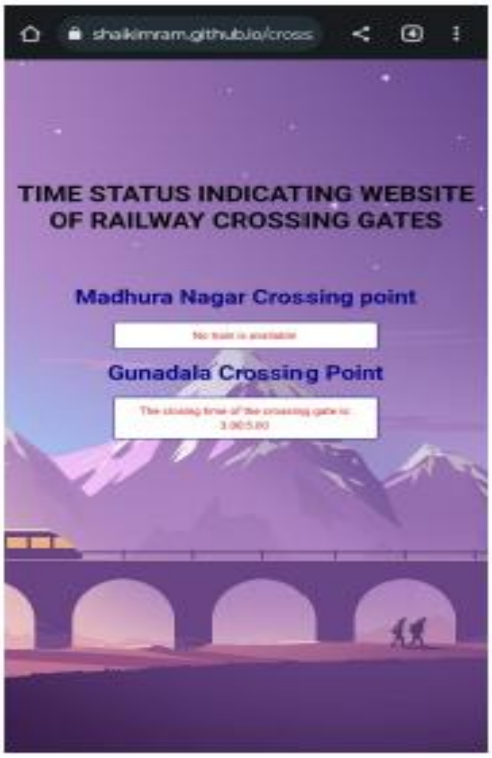

Figure 5: The time at which the gate is scheduled to be closed is displayed in URL through GITHUB

Commuters can plan their journeys more effectively, reducing the risk of being caught in unexpected railway delays, while railway authorities gain the tools they need to monitor and manage crossings efficiently. Furthermore, using GitHub as the hosting platform ensures the reliability and robustness of the system. GitHub is known for its security measures, data backup capabilities, and collaborative features, which makes it an ideal choice for hosting such critical railway data. Additionally, its accessibility and ease of use allow for seamless updates and maintenance of the system, ensuring that the displayed information remains accurate and up-to-date..

In summary, the integration of GitHub into the Time Status Indicating System for railway crossing gates is a forward-looking approach that leverages technology to enhance safety, convenience, and overall railway operations. By providing a user-friendly interface for displaying gate closure times, this system contributes to a more efficient and safer railway network.

VII. ADVANTAGES

The Time Status Indicating System for railway crossing gates, which displays scheduled gate closure times through a GitHub-hosted URL, offers several notable advantages: Enhanced Safety: One of the primary advantages of this system is improved safety. Commuters can easily access real-time information about when the railway crossing gates are scheduled to close, allowing them to plan their journeys accordingly. This reduces the likelihood of accidents and near misses at railway crossings, as people are aware of the gate closure times.

Efficient Commute Planning: This system empowers commuters with timely information, enabling them to plan their routes and schedules more efficiently. They can avoid unnecessary delays and make informed decisions about their travel routes, which contributes to smoother traffic flow. Real-Time Updates: By hosting the system on GitHub, updates to the gate closure schedule can be easily managed and pushed in real-time. This ensures that the displayed information remains accurate and up-to-date, even in the event of schedule changes due to unforeseen circumstances. Accessibility: The web-based interface provided by the GitHub-hosted URL makes it accessible to a wide range of users, including those with smartphones or computers. It is user-friendly and does not require any specialized software or hardware, making it inclusive and widely accessible. Cost Efficient: Leveraging GitHub for hosting eliminates the need for building and maintaining a separate infrastructure for displaying gate closure times. This cost-effective approach is especially beneficial for smaller railway operators or regions with budget constraints. Centralized Data Management: GitHub's version control and data management capabilities ensure that the gate closure schedule data is securely stored and easily accessible for authorized personnel. This centralized approach simplifies data management and reduces the risk of data loss or inconsistency.

Scalability: The system can be easily scaled to accommodate additional railway crossings or expanded functionality. As the railway network grows, this scalability ensures that the system can adapt to changing needs and requirements.

Open Source Collaboration: GitHub encourages open-source collaboration, allowing multiple contributors to improve and expand the system. This collaborative aspect can lead to innovations and enhancements that benefit the entire railway community.

Data Analytics: The system can potentially collect data over time, which can be analyzed to identify trends, optimize gate closure schedules, and improve overall railway operations. This data-driven approach can lead to further efficiency gains.

VIII. CONCLUSION

The development and implementation of a Time Status Indicating System for railway crossing gates represent a significant advancement in ensuring the safety and efficiency of railway operations. This innovative system has been designed to address the challenges and risks associated with level crossings by providing real-time information to both railway personnel and road users. The Time Status Indicating System enhances safety by reducing the likelihood of accidents and collisions at railway crossings. By offering a countdown or status display of when the crossing gates will open or close, it gives motorists and pedestrians ample time to prepare and make informed decisions. Furthermore, this system aids railway authorities in effectively managing their operations. It allows for better coordination of train schedules, maintenance work, and other activities related to the crossing gates. By incorporating advanced technologies such as sensors, communication networks, and automated controls, the system can adapt to changing circumstances, weather conditions, and emergencies, ensuring the continued safety of all stakeholders.

REFERENCES

[1] Ms. J. Praiseline Karunya M.E, "Automated Railway Gate Control and Object Detection using Wireless Communication" International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 NCIECC - 2017 Conference Proceedings.
[2] Atul Kumar Dewangan, Meenu Gupta, Pratibha Patel, "Automation of Railway Gate Control Using Microcontroller" International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 3, May 2012.
[3] M.Abinaya,Vidya, Thenmozhi "Automation of Railway Gate using Internet of Things (IoT) "International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Confcall 2018 Conference Proceedings.
[4] Chandolu Yeshwanth Sai Vivek, Dharaa C, Prakash P, "Railway Gate System: Railway Gate Status Detection" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878 (Online), Volume-7, Issue-6, March 2019
[5] N.Pavithra, K.Tamil selvi, M. Kowsalya, "Railway Track Monitoring and Accident Avoiding System" International Journal of Engineering Research & Technology(IJERT)ISSN: 2278-0181 ETEDM - 2019 Conference Proceedings.
[6] Poornima Mahesh, Mahesh Ambekar, Satya prakash Pandey, "TRAIN TRACKING SYSTEM BASED ON GPS & GSM" International Journal of Engineering Research & Technology (IJERT) ICONECT' 14 Conference Proceedings.
[7] Muhammad, Watit Benjapolakul, "Automated Railway Crossing System Using ZigBee" International Journal of Applied Science and Technology • December 2017.KMUTNB Int J Appl Sci Technol, Vol. 11, No. 1, pp. 15–19, 2018.