Railway Track Management System based on IoT devices and Software Application facilitating touch user interface

Dr. Kakoli Banerjee
Associate Professor
Department of Computer Science and Engineering, JSS Academy of Technical Education, Noida, India

Saurabh Ranjan, Naman Jain, Aayush Jaiswal, Harshit Verma, Divya Khanna, Prateek Mittal, Prakhar P. Bhardawaj

Department of Computer Science and Engineering, JSS Academy of Technical Education, Noida, India

I. ABSTRACT

The railway station deals with more than 12k trains daily which demand extremely fast and efficient control system. The manual control is slow, tedious, stressful and doesn't involve any holistic operational management. Hence we require an automated system to record and maintain different components of the railway track management system. In this paper, I propose the automation of three different areas of management of railway tracks: Monitoring of speed of the train, Temperature monitoring, and Lubrication. The basic idea of the execution of the project is, first of all, placing the sensors that help in storing required data such as rail temperature, the speed of the train, and then fetching the signals on Raspberry Pi and finally performing the computation. If the result of the computation goes beyond the specified limit then the buzzer will go off and nearby stations will be informed. For the lubrication system, the railway tracks need to maintain a fixed range of friction in turns with more than 3 degrees of turn i.e. 0.25 to 0.3u.[9] This is achieved by continuous lubrication system sensing the train wheels to pass by. This lets the train to pass the turn with maximum efficiency and least wear and tear. The monitoring of rail temperature helps in tackling the issue of rail buckling and any type of wear and tear due to the interaction of rail wheels with the tracks.

II. INTRODUCTION

Railway track management is an important task for the railway authorities as damaged tracks can cause mass destruction. Railway tracks are damaged due to excessive heat and pressure so it is important to inspect the tracks on a daily basis to avoid any hindrance.

Right now the inspection of the railway tracks is done manually by the engineers which are hectic and is not efficient. It consumes much time and manpower. To avoid this we came up with the Railway Track Management System by which the changes in the tracks can be monitored

Easily and more efficiently. Also, the repairing of the track can be done on time.

By using The Internet Of Things (IoT) and Rasberry Pi we can monitor the minute changes in the tracks like temperature, pressure, the speed of the passing train, etc. By computing the fetched data we can evaluate the condition of the track and hence the required action can be taken.

The Internet of Things (IoT), also sometimes referred to as the Internet of Everything (IoE), consists of all the web-enabled devices that collect, send and act on data they

acquire from their surrounding environments using embedded sensors, processors and communication hardware.

The Raspberry Pi is a low cost, small computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing and to learn how to program in languages like Scratch and Python.

Railway track management is a vast task, here we will be focusing on the three major tasks.

1. Temperature Monitoring
2. Speed of the passing train
3. Lubrication

All these tasks were done manually which is time-consuming and not much efficient but using sensors and raspberry pi it can be done more efficiently with minimal manpower.

The device along with the sensors has to be placed at some distance on the track which will fetch the data from the sensors and perform the calculations and hence the result will be displayed on the screen at the control room and hence according to the output the required action can be taken.

Fig 1. Shows the buckling of rail tracks

III. LITERATURE SURVEY

The proper management system has great importance in railways and to know the happenings on track and the issues occurring in the functioning of the railways it is important to monitor the factors affecting the system. As discussed heavily in [1,2,8] the requirement for a proper management system is inevitable. The tracks are affected by high temperature and the most common issue is buckling of tracks (Fig.1) and derailment. This is elaborated in a paper by R.S. Poonia [9]. The derailment of the train is an all cost avoidable scenario and to achieve the security the trains are being run with a proper monitoring and signaling system.

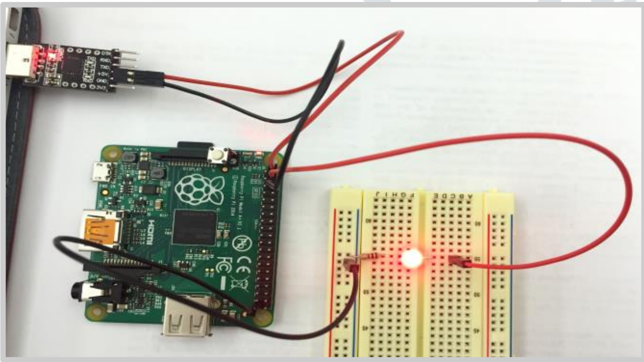


Fig 2. Raspberry Pi with the connections and breadboard

Fig 3. DS18b20 temperature sensor

IV. APPROACH

To fetch the temperature of the track we used DS18B20 temperature sensor [fig. 3], and then this temperature data is transferred to Raspberry Pi [fig. 2] through GPIO pins for further computation.

Fig 4. Proximity Inductive Sensor

This data is also used during the lubrication of the train.

To measure the speed of the train we used two proximity sensors [Fig.2] which can detect the metal wheels of the train. These two sensors were placed at some fixed distance, so we can detect that when the wheel passes both the sensors and by this data and distance between the proximity sensors we can calculate the speed of the train.

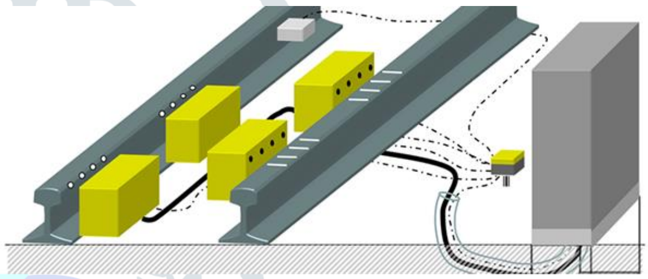


Fig 4. Lubrication System and pump

Next, we have to manage the lubrication of the train while the train is moving on the track. For this, we used a proximity sensor to detect that when the train is passing through that lubrication point.
As soon as the train is detected, the lubrication system is activated and by using the pump the lubricant is spread on tracks which are carried by the wheels of the trains over the turns of the track.

V. IMPLEMENTATION

STEP1*:*

The process begins with connecting all the sensors with the raspberry pi with correct connection points such that it facilitates the communication between the two.

STEP *2:*

The sensors are used to fetch the data from the outside world and transferred to the pi for calculations and use. The sensors are placed at the correct locations to fetch the data.

STEP 3:

The sensor includes a ds18b20 temperature sensor and proximity speed sensor and a buzzer. The buzzer is blown when the safe limit of data is crossed.

STEP 4:

The inductive sensor senses the metal object every time metal is passed close to it and sends the data to pi by turning on a gpio pin which is captured in the program. Likewise, the temperature sensor senses the temperature and writes to a file which is read in the program.

STEP 5:

Then the data fetched is processed upon to infer speed and temperature and then stored in the database along with mapping to other tables with railway information.

STEP 6:

The data is made visible in the GUI with touch input designed for best efficiency and smoother results. The flow is designed such that it is natural and yet diversified with options.

STEP 7:

The GUI is made on electron framework. Electron is an open source library developed by GitHub for building cross-platform desktop applications with HTML, CSS, and JavaScript. Electron accomplishes this by combining Chromium and Node.js into a single runtime and apps can be packaged for Mac, Windows, and Linux.

STEP 8:

The Gpio pins interact with pi using javascript and python. Inside electron on-off module of javascript is used to interact. GPIO access and interrupt detection with Node.js on Linux boards like the Raspberry Pi, C.H.I.P. or BeagleBone is provided by the on-off module.

STEP 9:

The package is then bundled in an executable file with “rimraf” bundler and code is uglified and minified to make is inaccessible to anyone else. This is done for the security of the code.

STEP 10:

Attaching sensors to the tracks. To attach sensors to the railway tracks we mounted them with tough mechanical containers. And fixed them to tracks.

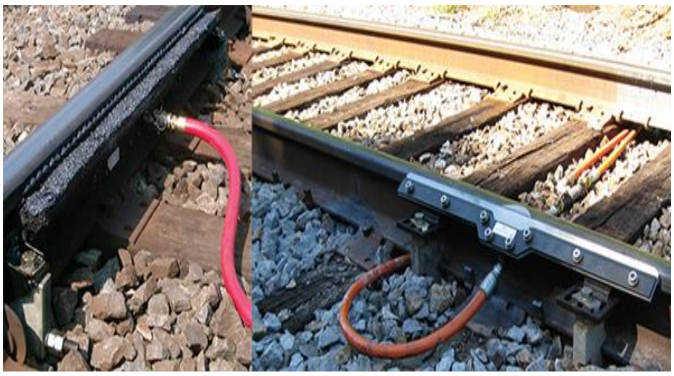


Fig 5. Pump and lubrication hose attached to the tracks

STEP 11:

The package as a whole work and connects with all the components with close bindings of software and hardware.

VI. WORKING

All the essential components such as pi, sensors, etc are connected together for the proper functionality to get accurate results.

Accuracy was the main goal to achieve that how much the data received is accurate and is enough to make the necessary calculations.

Fig 6. Raspberry Pi has 20 GPIO Pins

It was observed that sometimes the data received was highly changed in comparison to the previous result recorded.
The instant jump or fall was easily noticeable such as at time x, the recorded speed was 50 kmph and at the x+1 time the recorded speed went up to 95 kmph and at the x+2 time, 52 kmph was the recorded speed.
So it was clear that it is impossible for any train to reach from 50km/h to 95 km/h in a second.

Same goes for the temperature as:

- at x time - 46 degree Celsius
- at x+1 time - 98 degree Celsius
- at x+2 time - 44 degree Celcius

which clearly stated some issue in our system.

Although if all these abruptly increased or decreased records were ignored the rest of the results were enough to do the necessary calculations as these wrong calculations were found once in 100 records still those abruptly increased/decreased records were a concern.

Another thing was using these devices at various locations and collecting all those data at a particular server.
All the remote location was having their seperate devices and the data was collected separated on a local server on a MySQL database.

Thus it was needed to make the data centralized as to connect all the records at one server.

A centralized database (sometimes abbreviated CDB) (Fig 7) is a database that is located, stored, and maintained in a single location. This location is most often a central computer or database system, for example, a desktop or server CPU, or a mainframe computer.

Using Centralised database here: All the local server were connected to a centralized CRIS(Centre For Railway Information Systems) server which is the server used by IRCTC(Indian Railways Catering and Tourism Corporation) whose headquarters is situated at New Delhi.

So once a record for a particular track is completed at a local server the data is sent over the main server where the data is collected and categorized accordingly.

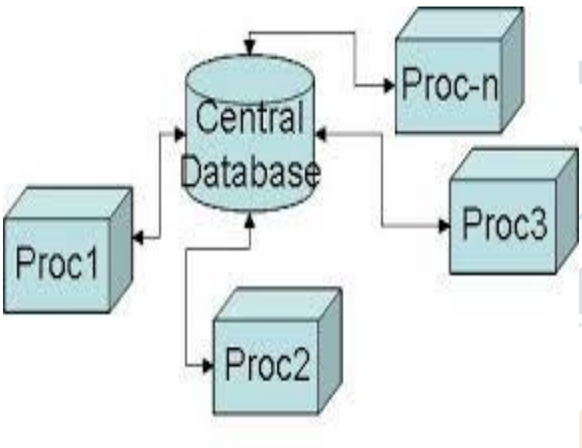
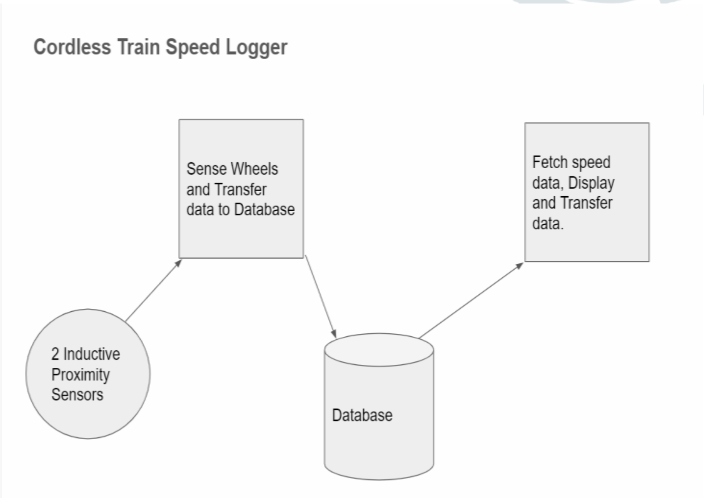


Fig 7. Centralized database system

Advantages of Centralized Database Management System is the **data integrity** is maximized as the whole database is stored at a single physical location. This means that it is easier to coordinate the data and it is as accurate and consistent as possible. The data redundancy is minimal in the centralized database.



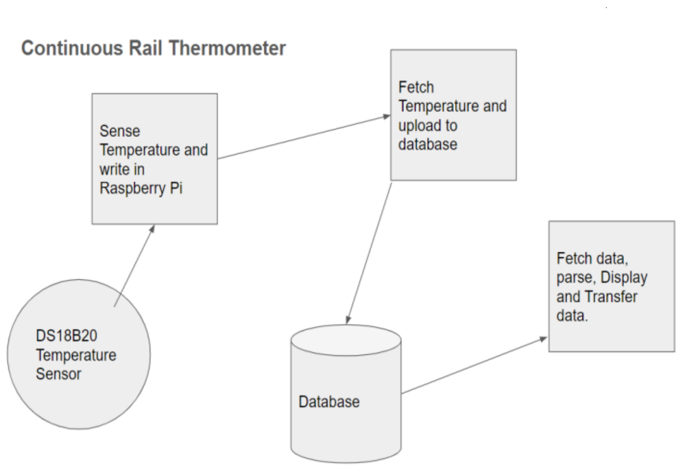


Fig 8. Dataflow diagram for all three modules

So any superuser activity can be directly done at the main server.
When the data from the local server to the main server the data is reviewed whether it's correct or not, once verified it is sent over the main server.

Another advantage was any glitch in any local server cannot directly affect the main data stored at the main server as it is filtered at a local level before sending over.

VII. RESULTS

All three modules are successfully installed and are capable of achieving the work for which they are designed. First of all the speed sensor module senses the wheels of the train and calculate the speed via known distance between the two sensors.

Train	t1	t2	speed
1	1557254850122	1557254850125	23
1	1557254850156	1557254850145	22
2	1557254850134	1557254850156	54
2	1557254850167	1557254850145	56
3	1557254850124	1557254850156	78
3	1557254850156	1557254850134	79
4	1557254850123	1557254850123	92
4	1557254850189	1557254850165	91

TABLE 1. Data stored for speed recording

For the temperature sensor module, the device stores the temperature of track for every minute and along with temperature it stores the flag for min and max temp reached. The buzzer is also turned on if any of the extremities is reached. As shown in table 2.

Train	time	temperature	max/min
1	1557254850122	23	NA
1	1557254850156	34	NA
2	1557254850134	45	MAX
2	1557254850167	32	NA
3	1557254850124	29	NA
3	1557254850156	28	NA
4	1557254850123	5	MIN
4	1557254850189	-8	MIN

TABLE 2. Data stored for temperature sensor

For the lubrication module. The sensors sense the wheels of the train and after a certain threshold value starts the pump for a fixed amount of time. This lets the fixed amount of lubricant to come out of the hose. This lubricant is spread over the tracks via moving wheels. These devices are placed at the staring of the turn to achieve lubrication of the tracks in turns. Any turn with 3 degrees or more curvature needs to be lubricated for greater efficiency. The lubricant is carried away till 2kms from the wheels of the train.

VIII. CONCLUSION

Everything in this project is completely automated, from the expansion of the railway tracks(by the heat generated due to the friction between the railway tracks and the train) to the lubrication of the track. Labour and the number of man-hour removal have been the main and focused objective. Accuracy is also managed while creating this project and all the extreme cases are handled like technical failures and environmental issues such that we don't receive faulty values.

IX. REFERENCES

[1] http://www.railsystem.net/rail-wheel-lubrication/

[2] http://www.railsystem.net/buckled-rail/

[3] IS:104-1979 (Reaffirmed 2004): Ready mixed paint, brushing, zinc chrome, priming

[4] IS:2074-1992 (Reaffirmed 2002): Ready mixed paint, air drying, red oxide, zinc chrome, priming-Specification

[5] IS:9000(Pt.XI)-1983 (Reaffirmed-2007): Basic environmental testing procedures for electronic and electrical items: Part II Salt mist test

[6] IS:9000(Pt.XV)/Sec.7-1982 (Reaffirmed-2004): Basic Environmental Testing Procedures for Electronic and Electrical Items - Part XV: Sealing Test.

[7] IS:2932-2003: Enamel, Synthetic, Exterior：

- Undercoating
- Finishing - Specification

[8] http://networkrail.co.uk/

[9] https://drive.google.com/file/d/1tKiPztLoUMjAStUmfeiUobq9VZpuJLCy/view?usp=sharing