Development of a Smart Railway System for Bangladesh

Masharul Bin Mahfuz*, Zohair Mehtab Ali, Md. Shakhawat Hossain, Avijit Das
Department of Electrical and Electronic Engineering.
BRAC University,
Dhaka, Bangladesh.
masharul92@gmail.com*

***Abstract*— This paper suggests an approach towards an intelligent and automatic management of a railway transportation system in order to prevent hazards like collisions and derailments. The system is designed as such that it is most viable for Bangladesh, however, it can also be easily implemented at any other railway infrastructure. The system features active train detection using global navigation satellite system (GNSS) coordinates and obstacle detection at level crossings using long range infrared; automatic signaling and gate control at level crossings using light emitting diodes (LEDs) and servo motors; automatic and manual communication between trains and level crossings using global system for mobile communication (GSM) technology and lastly development of a web-based central control system to monitor locations and activities of trains using navigation technology and to communicate with the entire railway community as well as the country's emergency services. Our design includes the integration and interaction of three separate sub-systems: central control system, level crossing system and train system. Implementation of such a system in Bangladesh Railway will not only provide a comprehensive level of safety in railway transportation but also take Bangladesh a step forward towards the much-anticipated dream by Bangladesh government of creating a 'Digital Bangladesh'.**

Keywords*— *Automated Railway Signalling, Train Monitoring, Intelligent Level Crossing, Automated Railway Control, Web Application, Obstacle Detection, GSM, GPRS, GNSS.

I. Introduction

Although railway has been in existence for less than 250 years, the opening of Liverpool and Manchester Railway in September 1830 and implementation of human based signaling can be considered as the beginning of the 'Railway Signaling age' [1]. The first usage of electricity in railway signaling system was made in the late 19th century. It was the installation of an electro-mechanical system by the Great Western Railway that would sound a horn to inform the train driver about an obstacle or a bell to inform that the passage was clear.

Among different type of signaling system, semaphore-signaling system is a traditional form of railway signaling. In this system, each signal is assembled with an arm like the one object attached to the mast. The movement of the object through different positions and angles indicate different signals [2]. The implementation of track circuit signaling system during 1875 was the first electrical means of automatic train detection. In this system section of a rail, track is attached to a transmitter at one end and receiver at the other. A train interrupts the flow of electricity from the transmitter to the receiver and hence informs about its presence [1, 3]. A similar form of train detection system is done with treadles and axle counters. These forms of train detection is termed as passive train detection as the train did not play any part in detection except for being present. radio based train control (RBTC) system is a new form of train detection also termed as active train detection (ATD) as the equipment of train onboard played a vital role in train detection. The European Rail Traffic Management System is a noteworthy example of such train detection system [4, 5]. With the passage of time, high-speed trains and more complex railway infrastructure were invented in order to increase the efficiency of means of communication [6]. So, traditional forms of train detection are eventually becoming obsolete. The latest implementation of automated train control system is Positive Train Control (PTC). The main objectives of the system are to track occupancies through centralized route and interlocking logic, issue movement authorities via wireless data links, preferably mobile communication to trains and work vehicles, real-time detection of trains' position, enforcement of automated and pre-configured speed limits and authorization of movement for trains [4, 7].

The two major aspects of railway hazards are train derailment and train collision [8]. In 1984, Bangladesh Railway took a step forward towards modernization by laying optical fiber based digital telecommunication network. However, semaphore and color signaling system is still a major form of railway signaling. Microprocessor based software controlled interlocking system was introduced for Ishurdi-Jamtoil railway section whose total transaction cost was approximately EUR 8,493,474 [3, 9]. However, railway accidents have been increasing at an annual rate of 16% mainly due to derailment and sometimes due to human error, which is unintentional but a common phenomenon [2, 10]. The danger due to a presence of high voltage in track circuits, possibility of leakage of current, railway mechanical failure due to passing of train overtime, etc. are a few limitations of the existing implementation of track circuit based train detection [3, 11].

In this paper, we have described a system, developed by us, which attempts to solve the problems of train derailment and train collisions by means of automated active train detection using short message service (SMS), GSM and GNSS based technology in order to increase the reliability of means of communication [12-14]. In order to enhance railway performance and safety requirements, we have integrated active train detection and obstacle detection at level crossings; automatic signaling and gate control at level crossings; automatic communication between trains and level crossings and lastly development of a web-based central control system

978-1-5090-1134-6/17/$31.00 ©2017 IEEE

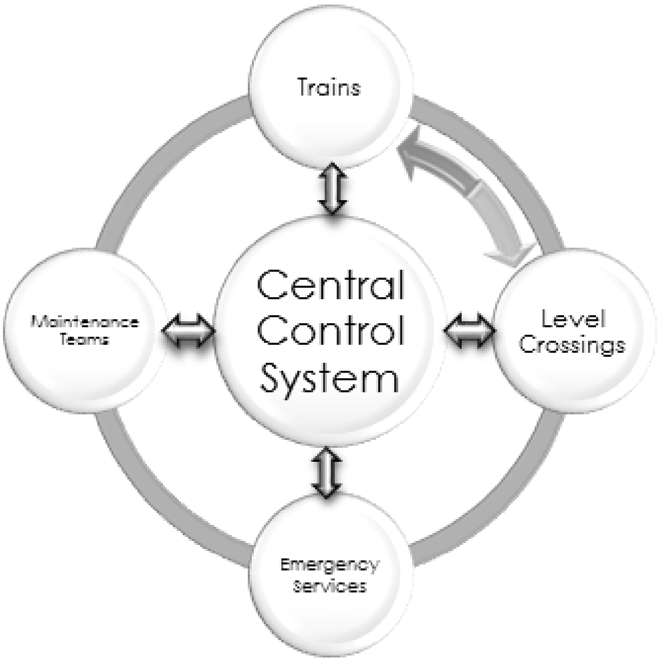

Fig. 1. Communication flow diagram of our system

to monitor locations and activities of trains using GNSS technology and to communicate among the entire railway community as well as the country's emergency services.

II. System Architecture

Our design is an integration and interaction of three other sub-systems: central control system, train system and level crossing system. Central control system is a web application where all railway activity can be monitored and controlled. It also contains all the relevant information about the railway system such as schedules, details of maintenance teams and emergency services. The train system and the level crossing system are devices containing microcontrollers, liquid crystal displays (LCDs), keypads, Simcom SIM808 modules, LEDs, switches and other output components.

Fig. 1 shows the communication flow of the entire system. Communication is done by means of GSM and general packet radio service (GPRS) using Simcom SIM808 module. The SIM808 module is a quad-band GSM and GPRS module, which combines GNSS technology for satellite navigation [15-16]. The module is integrated into our system for obtaining GNSS data from the satellites using the 1575.42 MHz GNSS antenna attached with it, uploading the data to the web server using GPRS and sending and receiving SMS using GSM technology. It is connected using the transistor–transistor logic (TTL) pins to the microcontroller and uses RS232 mode of communication [17]. The instructions used to control the SIM808 module are called attention (AT) commands [17]. We used an AT commands manual in order to determine the functions of each command [18-19]. Train system communicates with the central control system and the level crossings. Level crossings communicate with the central control system and trains. Central control system communicates with the emergency services of the country and vice versa. We used two methods of communication. One is via SMS messages to and from the SIM808 modules, and another is hypertext transfer protocol (HTTP), three-way handshake. Communication to and from the central control system is done using the HTTP. Trains and level crossings communicate by SMS messages. The core features of the entire system are:

1. Train-to-train collision detections
2. Excessive train speeds detection and response
3. Live view of all trains on-route of the country in a Google Map API in the central control system.
4. Communicate and command trains through the central control system.
5. Detect approaching trains at level crossings and open or close level crossing gates automatically.
6. Detect obstacles at different positions of the level crossing and warn the approaching train about it.

A. The Central Control System

The central control system (CCS) is a web-based application with restricted user access, which is used to centrally manage the whole system. The application is based on a Linux (x86_64) operating system and Symfony 2.8 framework. The server we used is Apache, version 2.4.25. The web application also uses CPanel of version 62.0 (Build 17). Programming languages used in order to construct the application are HTML5, CSS3, PHP, SQL, JavaScript and JQuery.

The web application will only be accessible by registered employees who would have to provide a unique identification (ID) number in order to register into the system. As shown in Fig. 2, the web application allows authorized personnel to view live locations of all trains on a Google Map API, view details (speed, departing station, destination station, current location, latitude and longitude) of trains which are on-route, store and view communication that takes place among all three subsystems. The system additionally features to periodically inform each train about any nearby trains. Operators of the system will also be able to view and update information related to railway transportation like train schedule, etc. We have also included a chat room in the web application for real-time interaction between employees. All necessary information regarding railway transportation is stored in its database and can be used and viewed by authorized users. Complete automation of a system is not always safe as the calculations are purely based on mathematical information. Therefore, the CCS provides a facility to exert authorized manual influence over any train whenever required. For a fully automated railway system, this web application can act as a remote monitoring and controlling device; and for trains, which are manually driven like in Bangladesh, it can be used as a proper guidance system for the driver.

The application is fully interactive and therefore can be accessed by any authorized personnel from any location with any device connected to the internet like a laptop, mobile phone, etc. Portability of the CCS makes the means of communication easier as it can be accessed irrespective of the location of a person.

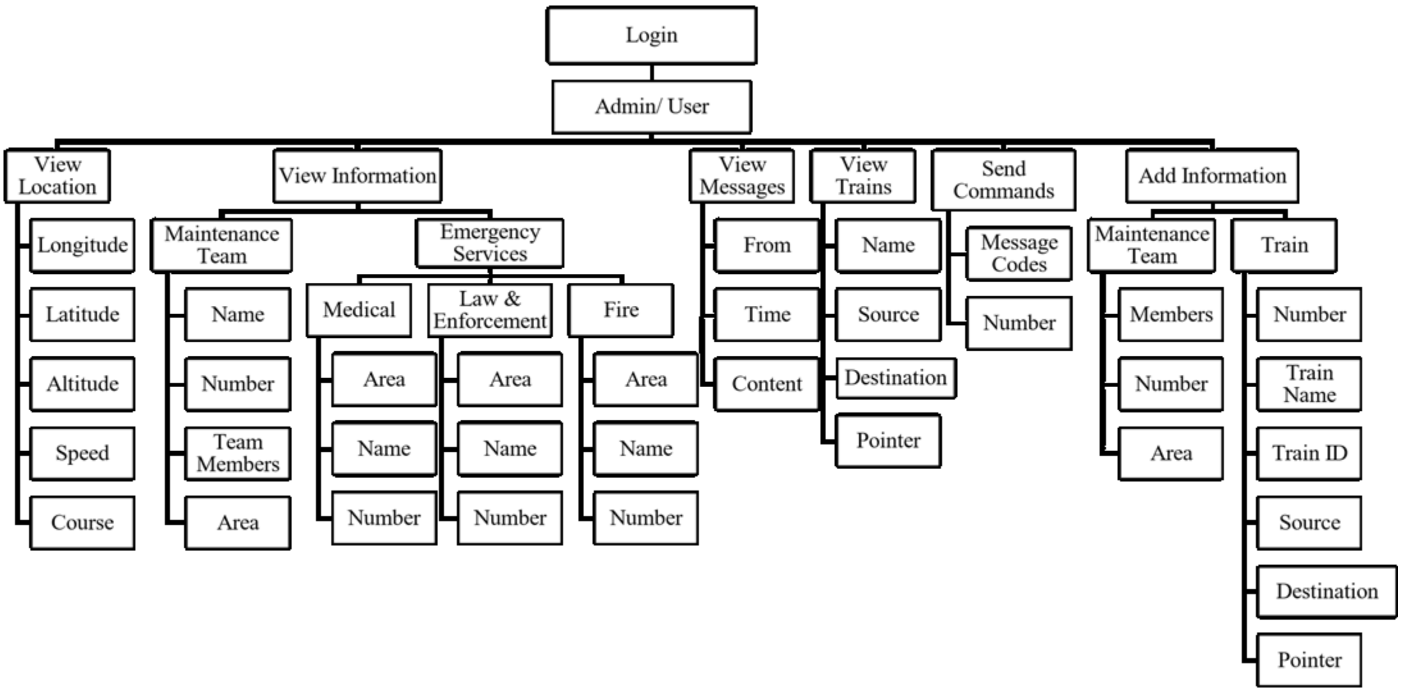

Fig. 2. Flow chart showing the features of the web application used in the central control system.

B. The Level Crossing System

The level crossing system comprises of one microcontroller unit that acts as the only core of the system. All other components are connected and controlled by this microcontroller as shown in Fig. 3. The most important tasks that the level crossing does in the whole process is to receive SMS via the SIM808 module from any incoming train and to check for obstacles on and around the tracks when a train is approaching towards it. It transmits appropriate data to the train about the road and track condition periodically until the obstacle has been cleared or the train leaves the level crossing region safely. It also opens or closes gates automatically based on the road and track condition and information provided by the incoming train. We have also kept an option of manual control over the gates of level crossing via a menu option.

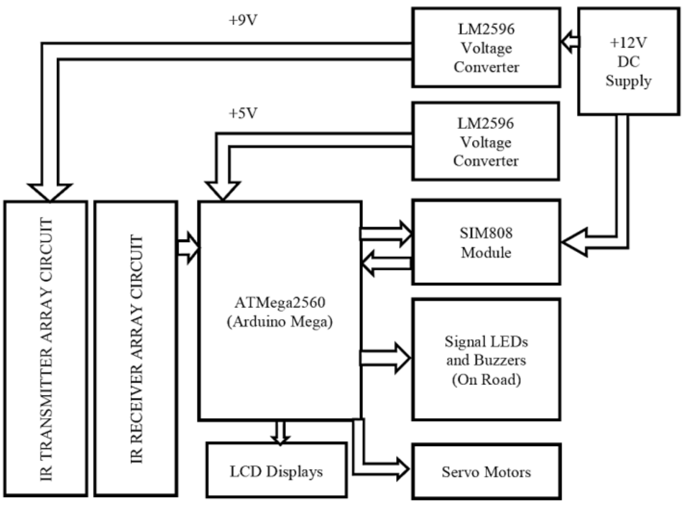

Fig. 3. Block diagram of the level crossing system

1. Obstacle Detection at Level Crossing

Obstacle detection is done using IR sensors. Normally IR transmitters transmit IR to a very short range like 2 meters at maximum. We designed a circuit which can transmit IR rays up to 10 meters range.

Fig. 4 shows the circuit schematic of the IR transmitter. We created an array of ten of these circuits due to ten positions of obstacle detection on the level crossing. The IR transmitter uses integrated circuit (IC) NE555 in 'Astable Multivibrator' mode of operation that uses the resistors R1, R2 and capacitor C1 to obtain an output frequency of 38 KHz [20-22]. The output frequency is set by changing the value of R1 using the formula in (1). We used BD140 (T1) which is a PNP transistor used to drive the infrared LED [23]. T1 is known as the Darlington high gain PNP transistor [24]. A resistor, R4 keeps the base of T1 HIGH in order to enhance the efficiency of IR transmitter.

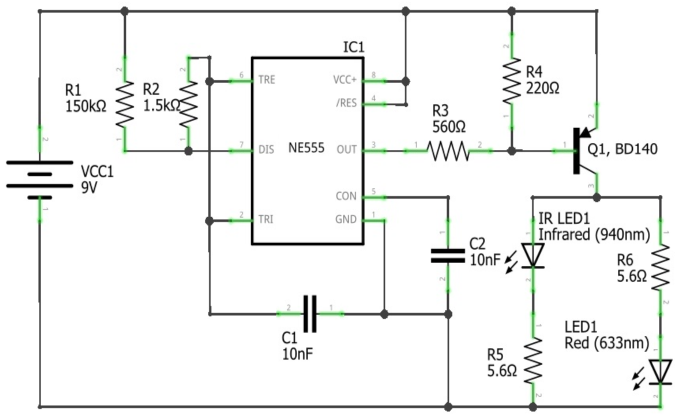

Fig. 4. Schematic of the IR transmitter circuit

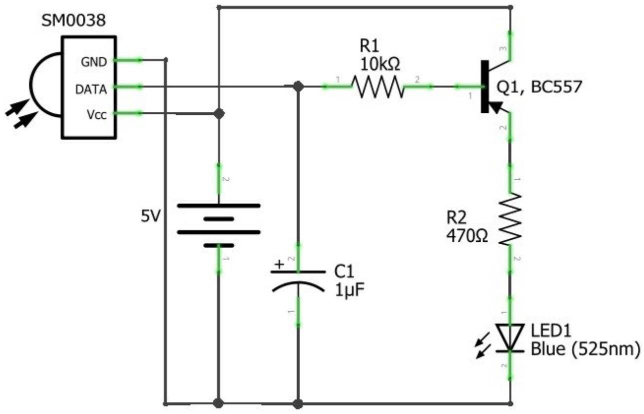

Fig. 5. Schematic of the IR receiver circuit

Fig. 5 shows the IR receiver circuit schematic. We used SM0038 IR receivers in the receiver circuit so that it receives only IR rays of frequency 38 kHz and no other rays [25]. This makes the system unique, safe and efficient as the obstacle detection process will not be affected by any external factors. The IR receiver circuit's output pins are connected to the analog pins of the microcontroller to use the values. We used BC557 PNP transistor to reverse the effect of the receiver, which means that whenever the output is HIGH, the LED will be OFF, and whenever it detects any IR of 38 KHz the LED will be ON [26]. The 470Ω resistor is used as a current limiting resistor for the LED and the 10kΩ resistor is used for providing proper biasing to the BC557 transistor. We created an array of ten circuits for the ten positions in a level crossing as shown in Fig. 6. S1 to S10 indicates the IR receiver numbers. T1 to T10 indicates IR transmitter numbers.

The formula for calculating the frequency of NE555 timer circuit in astable mode is [20-22]:

$$f = 1.44/ (R_1+2R_2)\, C_1 \qquad (1)$$

In (1) R_1 and R_2 are the resistance values shown in figure 4. C_1 is the capacitance values shown in figure 4. f is the output frequency of the NE555 timer IC.

Since our IR receivers receive IR rays of frequency 38 KHz, we needed to transmit a signal of a similar frequency from the IR transmitters. Thus, we set the value of f to 38 KHz and the values of R_1 and C_1 to 680Ω and 0.01µF in (1) respectively and calculated the value of R_2.

$$38 \text{ X } 10^3 = 1.44/ ((680+2 \text{ X } R_2) \text{ X } 0.01 \text{ X } 10^{-6}) \qquad (2)$$

$$\Rightarrow \quad R_2 = (1.44 - 0.01 \text{ X } 10^{-6} \text{ X } 38 \text{ X } 10^3 \text{ X } 680)/\ 2 \text{ X } 0.01 \text{ X } 10^{-6} \text{ X } 38 \text{ X } 10^3 \qquad (3)$$

$$\Rightarrow \quad R_2 = 1554.74\Omega \qquad (4)$$

We used 1.5KΩ resistors, which is the closest to the obtained values available in the local market.

Obstacle detection process starts as soon as the level crossing system knows that a train is arriving. The microcontroller continuously reads the values from the IR receiver output. Usually, when it receives 38 KHz IR rays, the values stay at around 1000; and decreases to around 500 if there are any obstacles between the transmitter-receiver pair. If the microcontroller reads the presence of an object by any sensors, it waits for five seconds to confirm that the object is an obstacle before performing the next tasks. If the object is an obstacle the microcontroller immediately commands the SIM808 module to send an SMS code to the approaching train. We have set three types of SMS codes to be sent to the train they are: 'OBS-111', 'OBS-100' and 'OBS-000'. 'OBS-111' indicates that the obstacle is present on the train's track; 'OBS-100' indicates that the obstacle is present near the train's track; 'OBS-000' indicates that there are no obstacles present. SMS codes to be sent are chosen depending on the position of the obstacle. Updates regarding obstacle detection are sent to the train at an interval of every five seconds until the train leaves the level crossing. Gates on a particular side of the road will remain open if any obstacle is detected on that side. This is done to prevent the bar from falling onto any vehicles and damage them and also allows the vehicle escape the hazardous zone and reach a safe position. This phenomenon is described in Fig.7.

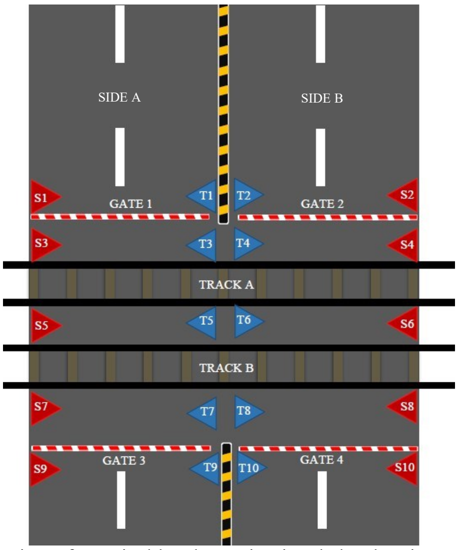

Fig. 6. Top view of a typical level crossing in Dhaka showing IR sensor positions for obstacle detection. (S1-S10 indicates IR receivers; T1-T10 indicates IR transmitters)

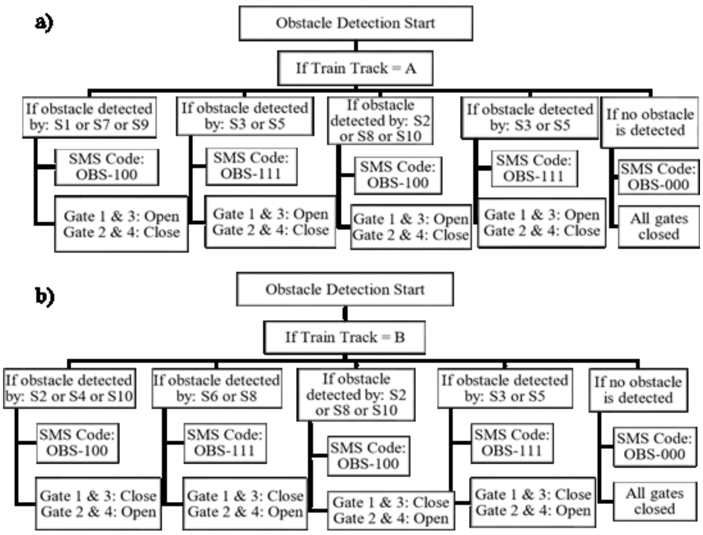

Fig.7. Flow chart showing SMS codes sent to train by the level crossing system and gate actions at the crossing a) if the approaching train is in track A. b) if the approaching train is in track B.

C. The Train System

The Train System is based on dual-core microcontroller architecture which uses two ATMEGA2560 microcontroller boards i.e. Arduino Mega, where one acts as a master and the other as a slave as shown in Fig. 8. Since these microcontrollers have single cores and single threads, they are not allowed to perform multi-tasking or any high-speed calculations [27]. Implementation of a dual-core system helped us to perform both tasks simultaneously and minimize time delays. The system contains four 20X4 LCDs to show all information to the driver.

The master and slave are connected to each other through Serial Data (SDA) and Serial Clock (SCL) pins. The method of communication between them is called I2C. This communication requires the use of Arduino's "Wire" Library. This library allows us to communicate with I2C devices [28].

1. Master

The primary tasks of the master Arduino are to read and process GNSS data from the SIM808 module attached to it. After the GNSS data is processed, it then transmits the data to the CCS using GPRS technology in the SIM808 module. Variables that are obtained and used from the GNSS data are, latitude, longitude, altitude, speed-over-ground, course-over-ground and UTC date and time. The GNSS data is also sent to the slave for further processing. The master Arduino is also responsible for sending predefined message codes via GPRS of SIM808 module to the CCS automatically. It also reads and displays the responses from the CCS, for example, information about a nearby train. A button named 'Start/Stop Journey' to start and stop a journey is also connected to the master. The train driver is only permissible to start a journey only if a positive confirmation is received from the CCS upon pressing the 'Start/Stop Journey' button. When the button is pressed, the master uploads all information regarding the journey to the CCS. Since then the train is able to communicate and upload locations to the CCS. This process is stopped as soon as the 'Start/Stop Journey' button is pressed again. The two LCDs connected to the master shows location information and journey status information respectively.

2. Slave

The prime function of the slave is to receive GNSS data from the master and detect any upcoming level crossing. Consecutively, it notifies any nearby level crossing about the train's arrival with formatted information via SMS message code using the SIM808 module. The GNSS co-ordinates of all level crossings and their corresponding SIM numbers are predefined in the slave's database from which it automatically detects a nearby level crossing within its predefined radius. On approaching any level crossing, the driver is automatically instructed to decrease the speed of the train below 80km/h due to safety. The slave Arduino also displays any incoming message from either the CCS or any nearby level crossing. It is also responsible for calculating the distance to the upcoming level crossing and performs the next action based on information core. The distance between two GNSS coordinates is done using the Haversine formula. Haversine formula is used to calculate the great-circle distance between two points – that is, the shortest distance over the earth's surface [29].

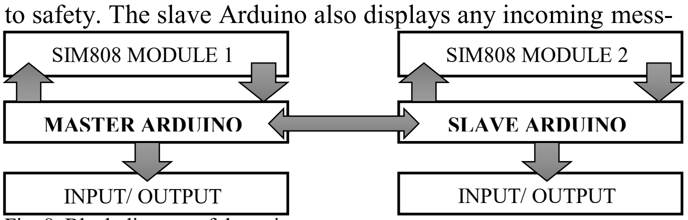

Fig. 8. Block diagram of the train system

Haversine formula:

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi 1 \cdot \cos\varphi 2 \cdot \sin^2(\Delta\lambda/2) \quad (5)$$

$$c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)}) \quad (6)$$

$$\text{Distance} = \text{Radius of the Earth in meters} * c \quad (7)$$

Here in (5), φ is bearing of latitude and λ is bearing of longitude. The slave includes an emergency alarm and a menu option in the display which is triggered by pressing a button connected to it. The two LCDs connected to the slave shows messages and instructions to the driver respectively.

Table I shows responses of the train system due to obstacle detection messages from the level crossing. Message codes 'OBS-111', 'OBS-100' and 'OBS-000' are used to indicate the presence of an obstacle at level crossings. On receiving these message codes, the slave automatically determines whether the train requires to slow down or not, by comparing with the

TABLE I. RESPONSES OF THE TRAIN SYSTEM DUE TO OBSTACLE DETECTION MESSAGES FROM THE LEVEL CROSSING

IF INCOMING MESSAGE = OBS-111			
Speed (Km/h)	**Distance (Km)**	**Instructions**	**LED**
60-80	0.0-1.2	Reduce Speed	Yellow
50-60	0.8-1.2	Maintain Current Speed	Green
	0.0-0.8	Reduce Speed	Yellow
30-50	0.4-1.2	Maintain Current Speed	Green
	0.0-0.4	Reduce Speed	Yellow
10-30	0.0-1.2	Maintain Current Speed	Green
60-80	1.0-1.2	Reduce Speed	Yellow
	0.8-1.0	Apply Brake level 1	Yellow
	0.6-0.8	Apply Brake level 2	Yellow
	0.4-0.6	Prepare to Stop	Yellow
	0.0-0.4	Stop	Red
50-60	1.0-1.2	Reduce Speed	Yellow
	0.8-1.0	Apply Brake level 1	Yellow
	0.6-0.8	Apply Brake level 2	Yellow
	0.4-0.6	Prepare to Stop	Yellow
	0.0-0.4	Stop	Red
30-50	1.0-1.2	Maintain Current Speed	Green
	1.0-0.6	Apply Brake level 2	Yellow
	0.4-0.6	Prepare to Stop	Yellow
	0.0-0.4	Stop	Red
10-30	1.2-0.4	Maintain Current Speed	Green
	0.2-0.4	Apply Brake level 1	Yellow
	0.0-0.2	Stop	Red
IF INCOMING MESSAGE = OBS-100			
60-80	0.0-1.2	Reduce Speed	Yellow
50-60	0.6-1.2	Maintain Current Speed	Green
	0.0-0.6	Reduce Speed	Yellow
10-50	0.0-1.2	Maintain Current Speed	Green
IF INCOMING MESSAGE = OBS-000			
<80	0.00-1.2	Reduce Speed	Yellow
>80	0.00-1.2	No Obstacles. Clear to Proceed	Green

current speed and by calculating the distance to the level crossing. The speed indications are represented by a message in the LCD and also by green, yellow and red LEDs connected to the slave. If the message code is 'OBS-111' the driver is gradually advised to stop the train until 'OBS-000' message is received from the level crossing. If the message code is 'OBS100', the driver is advised to reduce the speed of the train or maintain current speed.

III. Conclusion and Future Works

In a nutshell, this project has given us a unique experience of amalgamating the fields of electronics, communication and computer science, and has brought out a developed system that can be implemented in the real-life scenario. By implementing our system, train collisions can be predicted beforehand and necessary precautions could be taken in order to avoid any accidents. Derailments caused by excessive speeds could also be prevented as speeds of trains are always monitored in the CCS. The intuitive user interface of all three systems would allow users to combat emergency situations very easily. Automatic train detection and obstacle detection at level crossings would play a major role and guarantee a comprehensive amount of safety in the level crossings of Dhaka.

However, we look forward to incorporating additional features into our project in the near feature. On the first stage of the upgrade, we plan to create a web of wireless communication between the level crossing system and components (sensors, displays etc.) to ensure a seamless process. Our next upgrade will entail a cohesive obstacle detection system with the purpose of detecting obstacles at any point of the train track and a movement tracker in order to prevent the movement of a specific train through track switches in case they are in the wrong position. Our vision is also to make the system fully automated by implementing electronic braking system in the locomotives.

References

[1] Clark, S. "A History Of Railway Signalling (From The Bobby To The Balise)". IET Professional Development Course on Railway Signalling and Control Systems (RSCS 2012) (2012): n. pag. Web. 13 March 2017.

[2] "Analysis of Problems". Bangladesh Railway-Government of the People of Republic Bangladesh, April 20, 2017,

[3] de Bruin, Tim, Kim Verbert, and Robert Babuska. "Railway Track Circuit Fault Diagnosis Using Recurrent Neural Networks". IEEE Transactions on Neural Networks and Learning Systems 28.3 (2017): 523-533. Web. 10 March 2017.

[4] Palmer, J. "The Need For Train Detection". 11th IET Professional Development Course on Railway Signalling and Control Systems (2006): n. pag. Web. 10 March 2017.

[5] King, Karl M. "Systems Engineering Framework For Railway Control & Safety Systems". 2016 IEEE International Conference on Intelligent Rail Transportation (ICIRT) (2016): n. pag. Web. 10 March 2017.

[6] Siahvashi, Ali, and Bijan Moaveni. "Automatic Train Control Based On The Multi-Agent Control Of Cooperative Systems". The Journal of Mathematics and Computer Science 1.4 (2010): 247-257. Print.

[7] Kim, Junhyeong et al. "Mobile Hotspot Network Enhancement System For High-Speed Railway Communication". 2017 11th European Conference on Antennas and Propagation (EUCAP) (2017): n. pag. Web. 10 February 2017.

[8] Hartong, Mark, Rajni Goel, and Duminda Wijesekera. "Positive Train Control (PTC) Failure Modes". Journal of King Saud University 1.23 (2011): 311-321. Print.

[9] Badrun, Nico, Bart. "Railway Signalling and Interlocking in Bangladesh". ORET Evaluation 2007 ─ 2012 – Case Study of the ORET Transaction BD00023. (2015): 5-11. Web. 24 April 2017.

[10] Imtiaz, Abir et al. "Automated System For Bangladesh Railway". Global Science and Technology Journal 3.1 (2015): 25-40. Print.

[11] Davis, Chester Ross. "Signal System, Interlocking Plants, And Automatic Train Control On The San Francisco-Oakland Bay Bridge Railway". Electrical Engineering 59.3 (1940): 158-164. Web. 10 March 2017.

[12] Hoq, Mahbubul et al., "Design and Development of a Microcontroller Based Traffic Light Control System" Proceedings of BES Conference (2003): 122-126. Print.

[13] S. Y. Aye, "Design and Construction of LAN based Car Traffic Control System," Proceedings of World Academy of Science, Engineering and Technology 33 (2008): 627-630

[14] National Transportation Safety Board, "Implement Positive Train Control in 2015", Washington DC, 2015.

[15] Shanghai SIMCom Wireless Solutions Ltd, "SIM808_Hardware_Design_V1.02", Shanghai SIMCom Wireless Solutions Ltd, Shanghai, 2015. pp. 11-15.

[16] "SIM808 | SIMCom | smart machines, smart decision | simcom.ee", Simcom.ee. [Online]. Available: http://simcom.ee/modules/gsm-gprsgnss/sim808/. [Accessed: 10- Nov- 2016].

[17] "Interface GSM Module to Arduino - Send and Receive SMS", Electronic Circuits and Diagram-Electronics Projects and Design, 2006. [Online]. Available: http://www.circuitstoday.com/interface-gsm-module-witharduino. [Accessed: 25- Nov- 2016].

[18] Shanghai SIMCom Wireless Solutions Ltd., "SIM800 Series_AT Commands Manual_ V1.10", Shanghai SIMCom Wireless Solutions Ltd., Shanghai, 2016.

[19] Shanghai SIMCom Wireless Solutions Ltd., "SIM800 Series_GNSS_Application Note_V1.00", Shanghai SIMCom Wireless Solutions Ltd., Shanghai, 2017.

[20] T. van Roon, "Astable Operation", in 555 Timer Tutorial, T. van Roon, Ed. 1995.

[21] "555 Timer Operating Modes", 555-timer-circuits.com, 2017. [Online]. Available: http://www.555-timer-circuits.com/operating-modes.html. [Accessed: 12- Mar- 2017].

[22] Texas Instruments, "xx555 Precision Timers," NE555 datasheet, Sept. 1973 [Revised Sept. 2014].

[23] STMicroelectronics, "Complementary low voltage transistor," BD140 datasheet, 2008.

[24] K. Risona and M. Ogaba, "Design Of A 100w Power Amplifier With Darlington Complementary Symmetry Output Power Transistor", The University Of Nairobi, Nairobi, 2016.

[25] "SM0038 - TSOP1738 - 38KHz IR receiver", Rarecomponents.com. [Online]. Available: http://rarecomponents.com/store/1403. [Accessed: 17- Jan- 2017].

[26] Philips Semiconductors, "PNP general purpose transistors," BC557 datasheet, Mar. 1997 [Revised Apr. 1999].

[27] Atmel, "8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash," Atmel ATmega640/V-1280/V-1281/V-2560/V2561/V datasheet, 2014.

[28] "Arduino - Wire", Arduino.cc. [Online]. Available: https://www.arduino.cc/en/Reference/Wire. [Accessed: 28- Feb- 2017].

[29] C. Veness, "Calculate distance and bearing between two Latitude/Longitude points using haversine formula in JavaScript", Movable-type.co.uk, 2017. [Online]. Available: http://www.movabletype.co.uk/scripts/latlong.html. [Accessed: 12- Mar- 2017].