ISSN 2250-3153

# Intelligent Traffic Control System Based On Round Robin Scheduling Algorithm

**Lutfun Nahar Nipa*, Md. Mohibul Islam****

* Department of Mechanical Engineering, Rajshahi University of Engineering & Technology, Rajshahi-6204, Bangladesh.
** Department of Industrial & Production Engineering, Rajshahi University of Engineering & Technology, Rajshahi-6204, Bangladesh.

***Abstract-*** Traffic jam is one of the major problems in a densely populated mega city like Dhaka whereas its population and number of running vehicles are much more than its road capacity. Traffic signaling systems, inadequate manpower, narrow road spaces and overtaking tendency of drivers create pro-longed traffic jams. Due to traffic jam a substantial portion of working hours have to be left on streets which indirectly put adverse impact on economy and unavoidable road accident which results loss of lives. As the number of road users constantly increases, and resources provided by current infrastructures are limited. Intelligent traffic control system has become a very important issue. In this study based on Round Robin Scheduling Algorithm an automatic traffic control system is proposed. The main objective of this study is to reduce the overall waiting time of the vehicle at the cross junction point. For doing this a microcontroller is used which will make the drivers bound to follow the traffic rules by controlling traffic system that brings the result of decreasing the rate of accident, controlling crowd, lowering the tendency of road blocking etc. This approach can be applied in cross road junction which are so busy and the sectors those experience a great traffic load.

***Index Terms-*** Traffic control barricades; Round Robin Algorithm, Cross road junction

## I. Introduction

The growth of traffic jam in the road network of large cities in developing countries like Bangladesh is a serious concern in case of urban areas. The traffic jam at the road intersection is most crucial because the performance of intersection affect the overall productivity of the whole road network most significantly. To reduce conflicts and ensure orderly movement of traffic at the intersection generally different types of traffic control devices are used among which traffic control barricade is one of the most popular and effective controlling tool. Traffic control barricade is a sign and signal device which is used to guide and control traffic includes pedestrians, motor drivers by placing adjacent or over or along the cross road junction, highways and other public areas. It is mainly used to warn drivers and pedestrians for guiding in a work zone and to redirect traffic on high speed roads. In case of four way road it is really very much important to control traffic in a manageable way. Important cross road junctions of our country are Science lab junction, Shahabagh junction, Gazipur cross junction, Shapla Chottor, Motijhil etc. Traditional traffic light system is used at these cross junction which is not suitable at this present moment. Beside sometimes pedestrians and drivers would not like to follow traffic rule that causes unwanted incident. So authors feel interest to work on this problem area and authors think that there is a scope to develop a modern approach which is very conductive to reduce the present problem in traffic system. The major distinguishable aspect of proposed traffic control system is that after a predefined time the road will be automatically blocked and pedestrians and driver are bound to follow the traffic rule. Thus this real-time technique can be able to solve the problem. The system proposed here involves localized traffic routing for each intersection junction based on microcontroller networks. The proposed system has a central microcontroller at every junction which receives data from control room placed on the road. All programs are controlled by central programmable microcontroller. The Microcontroller makes use of the proposed programmed algorithm to find ways to manage and regulate traffic in a systematic manner efficiently.

The rest of the paper is organized as follows. Literature review is mentioned at section 2, research methodology is mentioned at section 3, working principle of the proposed system is discussed at section 4, mathematical model assumptions and its implementation is discussed in section 5, results and conclusion are mentioned at section 6 and 7 respectively. Finally references are mentioned at the last portion of this paper.

## II. Literature Review

Saeidi and Baktash (2012) mentioned that the process scheduling is one of the most important tasks of the operating system. One of the most common scheduling algorithms used by the most operating systems is the Round Robin method in which, the ready processes waiting in ready queue, seize the processor for a short period of time known as the quantum (or time slice) circularly. In their study a non-linear programming mathematical model is developed to determine the optimum value of the time quantum, in order to minimize the average waiting time of the processes [1]. Hussian, Sharma and Sharma (2013) mentioned that the most commonly used traffic controlling system in developing countries is the microcontroller based system. This system involves a predefined time interval setting for each junction road at an every junction [2]. Mahdi and Zuhairi (2013) mentioned that several accident cases on traffic control have been reported in past due to poor control of traffic control at cross roads. They focused that today's world speed is the ultimate word. Everyone is running a rat race and people definitely prefer to spend more time and utilize their energy in doing their respective professional and personal work rather than wasting both their valuable time and energy in commuting on road [3]. Yin et al (2005) introduced a distributed architecture of the intelligent control integrated system for area-wide incident response, information guidance based on signal control. They applied Multi-Agent technique to the system and introduces the relative cooperation and negotiation theories based on game theory, by which the system can realize real-time active intelligent control, especially to actively resolve the congestions happened or will happen at the intersection. They mentioned that coordination among several intersections one of the most important problems of the area control system. In order to realize the integrated control system, they presented the incident and congestion forecast algorithms. Finally, as one intersection signal Agent, it can realize the isolated intersection signal control strategy independently by means of fuzzy logic, which is also considering the bus-priority [4]. Bullock and Hendrickso (1994) mentioned that effective roadway control is hampered by a variety of organizational, financial and technical considerations. One major hurdle is the current reliance on outmoded field hardware and software. They introduced a computable language that can be used for constructing real time traffic control software. This computable language is designed to be configured by a graphical user interface that does not require extensive software engineering training to use, yet provides much more flexibility and capability then possible by simply changing program parameters. The model is based upon the function block metaphor commonly used for constructing robust and efficient real time industrial control systems. The software model has been implemented in C on an open architecture traffic controller (OATC) hardware platform and demonstrated under simulated conditions for applications such as signalized intersection control, ramp metering, and communications with existing traffic control devices [5]. Dakhole and Moon (2013) mentioned that traffic research has the goal to optimize traffic flow of people and goods. As the number of road users constantly increases, and resources provided by current infrastructures are limited, intelligent control of traffic will become a very important issue in the future. However, some limitations to the usage of intelligent traffic control exist. Avoiding traffic jams there are several models for traffic simulation. In their study they focused on optimization of traffic light controllers in a city using IR sensor and control traffic using ATMEGA 16 microcontroller [6]. Raheja, Dhadich and Rajpal (2012) mentioned that Round Robin Scheduling is designed for time-sharing systems. There are various CPU scheduling algorithms have been defined such as First Come First Served FCFS, Shortest Job First (SJF), Shortest Remaining Time Next (SRTN). They mentioned that all the decisions for the size of time quantum are usually based on the crisp parameters in the case of Round Robin Scheduling. But they claimed that sometimes in many cases these parameters may be vague or imprecise. In their study they introduced an algorithm to improve the performance of Round Robin Scheduling Algorithm considering imprecise parameters [7]. Raman and Mittal (2014) mentioned that CPU Scheduling is one of the fundamental concepts of Operating System. Round Robin (RR) CPU scheduling algorithm is optimal CPU scheduling algorithm in timeshared systems. The performance of the CPU depends on the selection of time quantum in timeshared systems. The time quantum taken in RR algorithm is static that decreases the performance of CPU. They focused selection of time quantum and proposed a new CPU scheduling algorithm for timeshared systems called as EDRR (Efficient Dynamic Round Robin) algorithm. Their objective is to make a change in Round Robin CPU scheduling algorithm so that the performance of CPU can be improved [8]. Darbari, Medhavi and Srivastava (2008) discussed the application of Petrinet as the workflow tool to model urban Traffic system. Their study is divided into phases: the first phase deals with orthogonal extension into phase: the first phase deals with orthogonal extension of Petrinet to enhance the permutation of control and traffic flow simulation. The second phase discussed application of continuous pertinent with intelligent agents to the model the UTS in continuous format with single central control agent [9]. Aye, Tun and Myo mentioned that toll collection systems commonly used in Myanmar is manual transaction. Hence they introduced a new method for toll collection system according to the weight of the vehicles. Toll gate collection system has been fabricated based on microcontroller purpose of collecting toll according to the weight of vehicle [10]. Throughout the literature review it is observed that researchers are trying to improve traffic control system by applying different tools and techniques. Someone has been tried to propose an automatic multicolor signaling system to maintain traffic rule at the cross road junction point. Some of the researchers have been tried to improved Round Robin Scheduling Algorithm performance by developing modified different algorithm. In this study authors are proposed an intelligent traffic control system based on Round Robin Scheduling Algorithm by using microcontroller that will automatically block the road after a predefined time interval and that bound to drivers to maintain the traffic rule at the cross road junction point.

## III. Research Methodology

At first present traffic problem at mega city of Bangladesh is practically observed. Then current traffic control system is critically observed to check its feasibility to solve the present problem in traffic control. Then some past research work regarding on traffic control system is studied. Finally a new approach is proposed to regulate the traffic control properly at Dhaka city of its different cross junction point. Round Robin Scheduling Algorithm is applied at this new approach in order to reduce the waiting time of moving vehicle at the cross junction point. Automatically running this proposed system microcontroller is used whose circuit diagram is mentioned below in fig. 3.1.

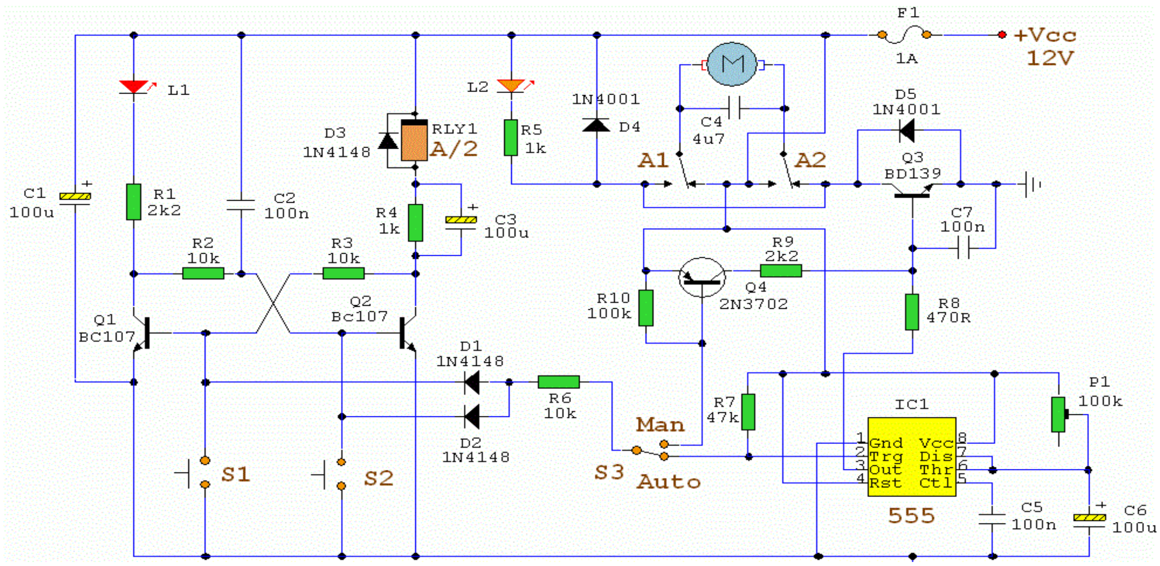


**Fig. 3.1: Circuit Diagram of a Microcontroller**

## IV. Working Principle

**Step 1 :** When the barrier G-1 is open & the rest barriers such as barrierG-2, G-3, G-4 are closed, only the vehicles from the road belong to barrier G-1 will be allowed to move in their required direction as shown in Fig.4.1 by the arrows & the vehicles of other roads belong to barrier G-2, G-3, G-4 are restricted to move.

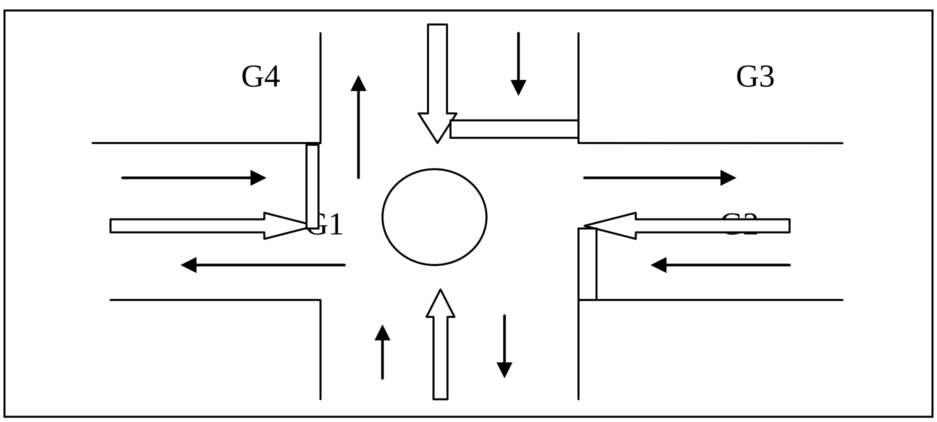


**Fig.4.1: Barricade of G1 is opened and G2, G3 and G4 are closed.**

**Step 2:** After a predetermined time period, controlled by microcontroller, barrier G-2 will open & barrier G-1, G-3, G-4 will be closed. So only the vehicles from the road belong to barrier G-2 will be moved in their required direction as shown in Fig. 4.2 by the arrows.

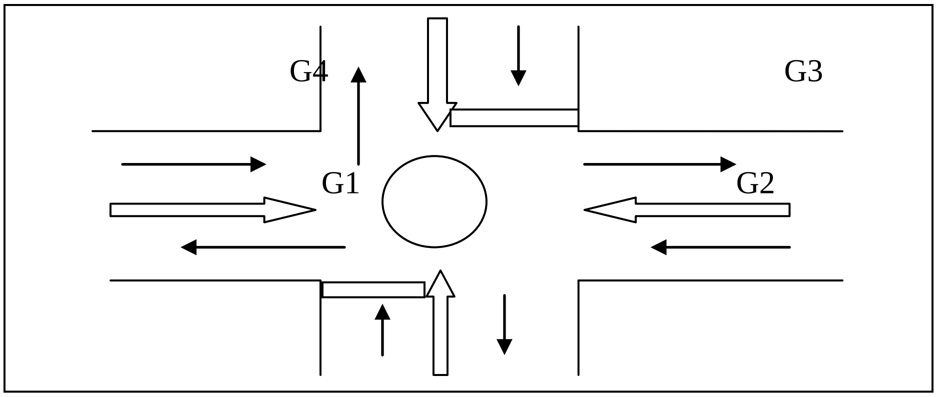


**Fig. 4.2: Barricade of G2 is opened and G1, G3 and G4 are closed.**

**Step 3:** Again after a predetermined fixed period barrier G-3 will be opened but barrier G-1, G-2 and G-4 will be closed. So only the vehicles from the road belong to barrier G-3 will be allowed to move in their required direction as shown in Fig.4.3 by the arrow

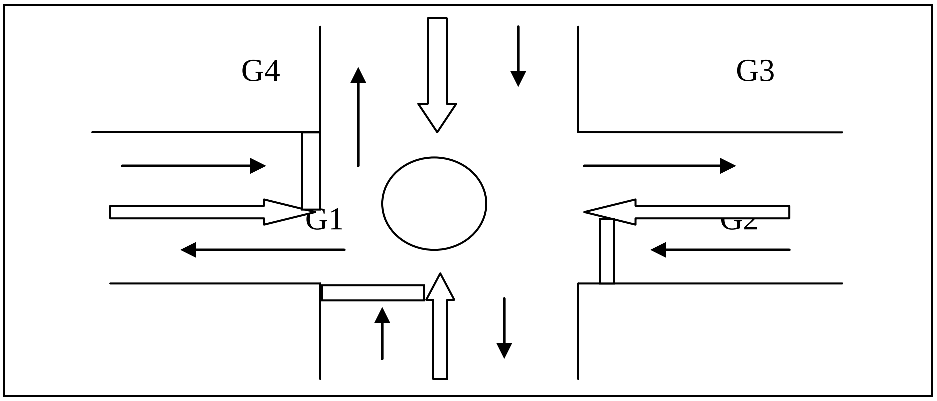


**Fig.4.3: Barricade of G3 is opened and G1, G2 and G4 are closed.**

**Step 4:** At the end on the cycle, barrier G-4 will be opened but barrier G-1, G-2, G-3 will be closed. So only the vehicles from the road belong to barrier G-4 will be allowed to move in their required directions shown in Fig. 4.4 by the arrows.

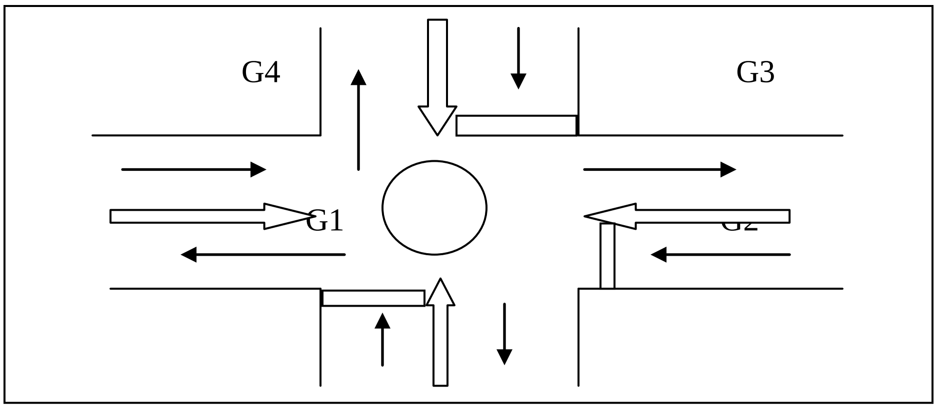


**Fig. 4.4: Barricade of G4 is opened and G1, G2 and G3 are closed.**

In such a way the total process will be repeated continuously until different kinds of requirements will be occurred. Pictorial view of simulated proposed traffic control system by using microcontroller is mentioned below in fig.4.5

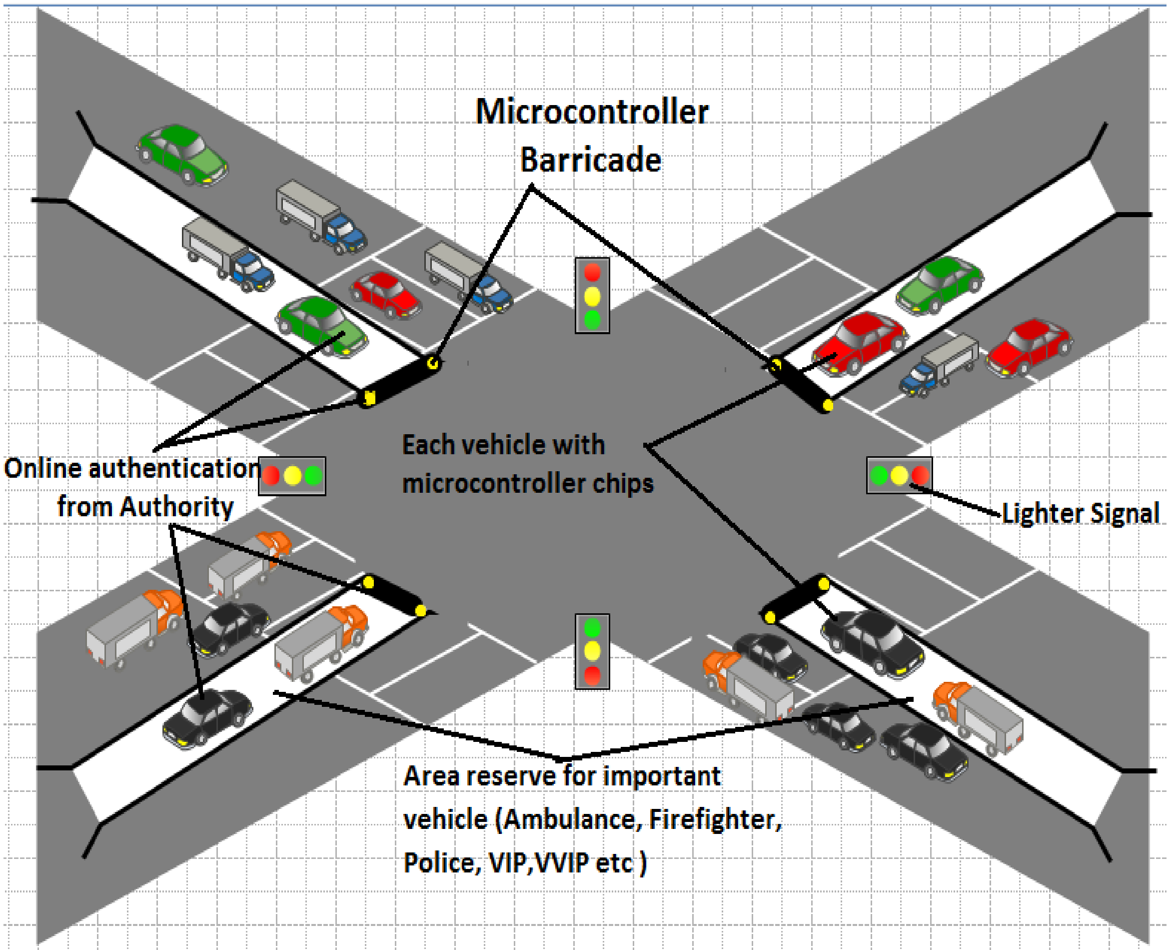


**Fig.4.5: Microcontroller Based Traffic Controlling Barricade System**

## V. Mathematical Model and its Implementation

### 5.1 Mathematical Model

Round Robin (RR) scheduling algorithm is used to determine the waiting time of proposed traffic control system. RR is the scheduling algorithm used by the CPU during execution of the process. RR is designed specifically for time sharing systems. It is similar to first come first serve scheduling algorithm but the preemption is the added functionality to switch between the processes. A small unit of time also known as time slices or quantum is set. The ready queue works like circular queue. All processes in this algorithm are kept in the circular queue also known as ready queue. Each new process is added to the tail of the ready/circular queue. By using this algorithm, CPU makes sure, time slices (any natural number) are assigned to each process in equal portions and in circular order, dealing with all process without any priority. It is also known as cyclic executive. The main advantage of round robin algorithm over first come first serve algorithm is that it is starvation free. Every process will be executed by CPU for fixed interval of time (which is set as time slice). So in this way no process left waiting for its turn to be executed by the CPU.

Pseudo Code:

CPU scheduler picks the process from the circular / ready queue, set a timer to interrupt it after 1 time slice / quantum and dispatches it. If process has burst time less than 1 time slice / quantum.

- Process will leave the CPU after the completion
- CPU will proceed with the next process in the ready queue / circular queue.

Else, if process has burst time longer than 1 time slice/quantum

- Timer will be stopped. It causes interruption to the Operating system.
- Executed process is then placed at the tail of the circular / ready queue by applying the context switch.
- CPU scheduler then proceeds by selecting the next process in the ready queue.

Here, User can calculate the average turnaround time and average waiting time along with the starting and finishing time of each process.

Turnaround time: It’s the total time taken by the process between starting and the completion

Waiting time: It’s the time for which process is ready to run but not executed by CPU scheduler.

### 5.2 Model Implementation

ISSN 2250-3153

Some assumptions are considered to implement the model which mentioned as follows:

- Intersection or Junction point is consisted of 4 roads and there is divider in every road.
- Roads are allowed for one way movement.
- Every Road is considered as Gate and this Gate assumed as a name of process. Such as Gate 1 is considered as process number 1, Gate 2 as process number 2 etc.
- Arrival time, Burst times are taken based on assumption considering real situation.
- Three Cases are considered which are as follows:

**Case 1:** When Arrival Time (AT) of vehicle along road is different.

**Case 2:** When Arrival Time (AT) of vehicle along road is zero i.e. initially vehicles are waiting on the road for passing the Junction point.

**Case 3:** When AT is considered as a fixed equal time for every road.

Now according to RR scheduling algorithm above Cases are executed below as follow:

**Case 1:**

Predefined time slice or quantum: 1 minute.

| Road no. or Gate no. | Arrival Time (AT) min | Burst Time or Service Time (min) |
|---|---|---|
| G1 | 0 | 2 |
| G2 | 1 | 3 |
| G3 | 2 | 2 |
| G4 | 3 | 2 |

Here in Round Robin, what will happen is as follow:

**<u>At 0 minute:</u>**

At 0 minute only one gate G1 is ready for open and others gates are remained closed. Gate G1 that has burst time is 2 minutes. So, request queue only G1 hence CPU will do 1 quantum predefined time 1 minutes for G1.

**<u>At 1 minute</u>**

At 1 minute vehicles are now waiting at G2 for passing the junction point that has 3 minutes burst time. Total burst time of G1 is 2 minutes from which 1 minute is completed at previous step and still it's has 1minute burst time. So the request queue is G2 and G1. At previous turn, CPU worked on G1 now CPU will work on G2 and G1 will be in request queue for further processing.

**<u>At 2 minute</u>**

At 2 minute vehicles are now waiting at G3 for passing the junction point that has 2 minutes burst time. Total burst time of G2 is 3 minutes from which 1 minute is completed at previous step and still it's has 2 minute burst time. But G1 is already in requested queue. So the request queue is G1, G3 and G2. At previous turn, CPU worked on G2, now CPU will work on G1and G3 and G2 will be remained in request queue for further processing.

**<u>At 3 minute</u>**

At 3 minute vehicles are now waiting at G4 for passing the junction point that has 2 minutes burst time. Total burst time of G1 is 2 minutes that has been completed at previous step and there is no burst time of it. But G3 and G2 is already in requested queue. So the request queue is G3, G2 and G4. At previous turn, CPU worked on G1, now CPU will work on G3 and G2, G4 will be remained in request queue for further processing.

**<u>At 4 minute</u>**

At 4 minutes vehicles now are waiting at each gate. Total burst time of G3 is 2 minutes from which 1 minute is completed at previous step and there is still 1 minute burst time is remained. But G2 and G4 is already in requested queue. So the request queue is G2, G4 and G3. At previous turn, CPU worked on G3, now CPU will work on G2 and G4, G3 will be remained in request queue for further processing.

**<u>At 5 minute</u>**

At 5 minutes vehicles now are waiting at each gate. Total burst time of G2 is 3 minutes from which 2 minute is completed at previous two steps and there is still 1 minute burst time. But G4 and G3 are already in requested queue. So the request queue is G4, G3 and G2. At previous turn, CPU worked on G2, now CPU will work on G4 and G3, G2 will be remained in request queue for further processing.

**<u>At 6 minute</u>**

At 6 minutes vehicles now are waiting at each gate. Total burst time of G4 is 2 minutes from which 1 minute is completed at previous steps and there is still 1 minute burst time. But G3 and G2 are already in requested queue. So the request queue is G3, G2 and G4. At previous turn, CPU worked on G4, now CPU will work on G3 and G2, G4 will be remained in request queue for further processing.

**<u>At 7 minute</u>**

At 7 minutes vehicles now are waiting at each gate. Total burst time of G3 is 2 minutes from which 2 minute is completed at previous two steps and there no burst time. But G2 and G4 are already in requested queue. So the request queue is G2 and G4. At previous turn, CPU worked on G3, now CPU will work on G2 and G4 will be remained in request queue for further processing.

**<u>At 8 minute</u>**

At 8 minutes vehicles now are waiting at gate G4. Total burst time of G2 is 3 minutes from which 3 minute is completed at previous three steps and there no burst time. Now only G4 is requested queue. So the request queue is G4. At previous turn, CPU worked on G2, now CPU will work on G4 and total burst time of G4 is completed through this step. In this way one cycle is completed. Total process is represented in under mentioned Gantt chart.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| G1 | G2 | G1 | G3 | G2 | G4 | G3 | G2 | G4 | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Therefore individual waiting time is G1: (2-1) = 1 minute, G2: (1-0) + (4-2) + (7-5) = 5 minute, G3: (3-0) + (6-4) = 5 minute, G4: (5-0) + (8-6) = 7 minute, So total waiting time = 1+5+5+7 =18 minute and average waiting time of each gate = 18/4 = 4.5 minute.

**Case 2:**

Predefined time slice or quantum: 1 minute.

| Road no. or Gate no. | Arrival Time (AT) min | Burst Time or Service Time (min) |
|---|---|---|
| G1 | 0 | 2 |
| G2 | 0 | 3 |
| G3 | 0 | 2 |
| G4 | 0 | 2 |

Here in Round Robin, what will happen is as follow:

**At 0 minute:**

At 0 minute four gates are ready for open but according to sequence gate will be opened i.e. requested queue are G1, G2, G3 and G4. G1 that has burst time is 2 minutes. CPU will do quantum predefined time 1 minute on G1.

**At 1 minute**

At 1 minute there is still remaining 1minute of burst time of G1 hence it will in queue after the G2, G3 and G4. So the requested queue G2, G3, G4 and G1. At previous turn, CPU worked on G1 now CPU will work on G2.

**At 2 minute**

At 2 minute there is still remaining 2 minutes of burst time of G2 hence it will in queue after the G3, G4, and G1. So the requested queue G3, G4, G1 and G2. At previous turn, CPU worked on G2 now CPU will work on G3.

**At 3 minute**

At 3 minute there is still remaining 1 minutes of burst time of G3 hence it will in queue after the G4, G1 and G2. So the requested queue G4, G1, G2 and G3. At previous turn, CPU worked on G3 now CPU will work on G4.

**At 4 minute**

At 4 minute there is still remaining 1 minutes of burst time of G4 hence it will in queue after the G1, G2 and G3. So the requested queue G1, G2, G3 and G4. At previous turn, CPU worked on G4 now CPU will work on G1.

**At 5 minute**

At 5 minute there is no busting time of G1 hence it will be out from the requested queue. Now the requested queue is G2, G3 and G4. At previous turn, CPU worked on G1 now CPU will work on G2.

**At 6 minute**

At 6 minute there is still remaining 1 minutes of burst time of G2 hence it will in queue after the G3 and G4. So the requested queue G3, G4 and G2. At previous turn, CPU worked on G2 now CPU will work on G3.

**At 7 minute**

At 7 minute there is no burst time of G3, hence it will be out from the requested queue. Now the requested queue is G4 and G2. At previous turn, CPU worked on G3 now CPU will work on G4.

**At 8 minute**

At 8 minute there is no burst time of G4, hence it will be out from the requested queue. Now the requested queue is only G2. At previous turn, CPU worked on G4 now CPU will work on G2. No there is bursting time of any gate. In this way one cycle is completed. Total process is represented in under mentioned Gantt chart.

| G1 | G2 | G3 | G4 | G1 | G2 | G3 | G4 | G2 |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Therefore individual waiting time is G1: (4-1) = 3 minute, G2: (1-0) + (5-2) + (8-6) = 6 minute, G3: (2-0) + (6-3) = 5 minute, G4: (3-0) + (7-4) = 6 minute. So total waiting time = 3+6+5+6 =20 minute and average waiting time of each gate = 20/4 = 5 minute

**Case 3:**

Predefined time slice or quantum: 1 minute.

| Road no. or Gate no. | Arrival Time (AT) min | Burst Time or Service Time (min) |
|---|---|---|
| G1 | 2 | 2 |
| G2 | 2 | 3 |
| G3 | 2 | 2 |
| G4 | 2 | 2 |

Here in Round Robin, what will happen is as follow:

**From 0 to 2 minute:**

At 0 minute four gates are closed and two minutes is waiting for vehicle arrival. But after two minutes four gates are ready for passing vehicles. So after 2 minute requested queue is G1, G2, G3 and G4. CPU will do quantum predefined time 1 minute on G1.

**At 3 minute**

At 3 minute there is still remaining 1minute of burst time of G1 hence it will in queue after the G2, G3 and G4. So the requested queue G2, G3, G4 and G1. At previous turn, CPU worked on G1 now CPU will work on G2.

**At 4 minute**

At 4 minute there is still remaining 2 minutes of burst time of G2 hence it will in queue after the G3, G4, and G1. So the requested queue G3, G4, G1 and G2. At previous turn, CPU worked on G2 now CPU will work on G3.

**At 5 minute**

At 5 minute there is still remaining 1 minutes of burst time of G3 hence it will in queue after the G4, G1 and G2. So the requested queue G4, G1, G2 and G3. At previous turn, CPU worked on G3 now CPU will work on G4.

**At 6 minute**

At 6 minute there is still remaining 1 minutes of burst time of G4 hence it will in queue after the G1, G2 and G3. So the requested queue G1, G2, G3 and G4. At previous turn, CPU worked on G4 now CPU will work on G1.

**At 7 minute**

At 7 minute there is no busting time of G1 hence it will be out from the requested queue. Now the requested queue is G2, G3 and G4. At previous turn, CPU worked on G1 now CPU will work on G2.

**At 8 minute**

At 8 minute there is still remaining 1 minutes of burst time of G2 hence it will in queue after the G3 and G4. So the requested queue G3, G4 and G2. At previous turn, CPU worked on G2 now CPU will work on G3.

**At 9 minute**

At 9 minute there is no burst time of G3, hence it will be out from the requested queue. Now the requested queue is G4 and G2. At previous turn, CPU worked on G3 now CPU will work on G4.

**At 10 minute**

At 10 minute there is no burst time of G4, hence it will be out from the requested queue. Now the requested queue is only G2. At previous turn, CPU worked on G4 now CPU will work on G2. Now there is no bursting time of any gate. In this way one cycle is completed. Total process is represented in under mentioned Gantt chart.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Waiting | Waiting | G1 | G2 | G3 | G4 | G1 | G2 | G3 | G4 | G2 |

0 1 2 3 4 5 6 7 8 9 10 11

Therefore individual waiting time is G1: (4-1) = (2-0) + (6-3) = 5 minute, G2: (3-0) + (7-4) + (10-8) = 8 minute, G3: (4-0) + (8-5) = 7 minute, G4: (5-0) + (9-6) = 8 minutes. So total waiting time = 5+8+7+8 =28 minute and average waiting time of each gate = 28/4 = 7 minutes.

## VI. Results

Throughout the analysis it is observed that the proposed new automatic traffic control system is practically feasible and applicable at Dhaka city at its different junction point. Under considered three cases such as Case 1, Case 2 and Case 3 it is found that total waiting time in total system for a specific cycle is 18 minute, 20 minute and 28 minute respectively and the average waiting time of individual gate is 4.5 minute for Case 1, 5 minute for Case 2 and 7 minute for Case 3. It is critically observed that Case 1 is better than other two cases and also observed that Case 1 is more feasible for practical implementation.

## VII. Conclusion

This project is based on a very effective way of optimizing traffic, with redefinition of threshold values for a real time application. This works to control traffic on four way roads according to traffic control barricades which is functioned by microcontroller. This proposed system will be able to build a developed country with less traffic jams and it will also help the emergency vehicle to reach in time to the destination. So, this intelligent system will help us to control traffic in more autonomous way.

## References


[1] Saeidi, S., and Baktash, H. A., 'Determining the Optimum Time Quantum Value in Round Robin Process Scheduling Method', *I.J. Information Technology and Computer Science*, 2012, vol. 10, pp.67-73.

[2] Hussian, R., Sharma, S., Sharma, V., 'WSN Applications: Automated Intelligent Traffic Control System Using Sensors', *International Journal of Soft Computing and Engineering (IJSCE)*, 2013, vol. 3, no. 3, pp. 2231-2307.

[3] Mahdi, A. S., and Zuhairi, A., 'Automatic Railway Gate and Crossing Control based Sensors & Microcontroller', *International Journal of Computer Trends and Technology (IJCTT)*, 2013, vol. 4, no.7, 2013, ISSN: 2231.

[4] Yin ZHU, Huapu LU, Junli Wang and Qiang LIU, 'Research on Urban Traffic Intelligent Control Integrated System', *Journal of the Eastern Asia Society for Transportation Studies*, 2005, vol.6, pp.2349 – 2364

[5] Bullock, D., and Hendrickso, C., 'Roadway Traffic Control Software', *IEEE Transactions on Control Systems Technology*, 1994, vol. 2 no. 3

[6] Dakhole, A. Y., and Moon, M. P., 'Design of Intelligent Traffic Control System Based On ARM', *International Journal of Advance Research in Computer Science and Management Studies*, 2013, vol. 1, no. 6, ISSN: 2321-7782 (Online).

[7] Raheja, S., Dhadich, R., and Rajpal, S., 'An Optimum Time Quantum Using Linguistic Synthesis for Round Robin CPU Scheduling Algorithm', *International Journal on Soft Computing (IJSC)*, 2012, vol.3, no.1

[8] Raman and Mittal, P. K., 'An Efficient Dynamic Round Robin CPU Scheduling Algorithm', *International Journal of Advanced Research in Computer Science and Software Engineering*, 2014, vol. 4, no.5, ISSN: 2277 128X.

[9] Darbari, M., Medhavi, S., and Srivastava, A. K., 'Development Of Effective Urban Road Traffic Management Using Workflow Wechniques For Upcoming Metro Cities Like Lucknow (India)', *International Journal of Hybrid Information Technology*, 2008, vol. 1, no.3

[10] Aye, Y.M., Tun, Z. M., 2 and Myo. H. T., 'Design and Construction of Toll Gate Control System Based On Microcontroller', *International Journal of Electronics and Computer Science Engineering*, vol. 3, no. 3, pp. 262-268.


## Authors


**First Author** – Lutfun Nahar Nipa, Department of Mechanical Engineering, Email: (Corresponding author) nahar_nipa@yahoo.com , Rajshahi University of Engineering & Technology, Rajshahi-6204, Bangladesh.

**Second Author** – Md. Mohibul Islam, Department of Industrial & Production Engineering, Email: mohibul05ipe@yahoo.com , Rajshahi University of Engineering & Technology, Rajshahi-6204, Bangladesh.