TELKOMNIKA, Vol.12, No.4, December 2014, pp. 787~794
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v12i4.497

Pre-Timed and Coordinated Traffic Controller Systems Based on AVR Microcontroller

Freddy Kurniawan[1], Denny Dermawan[1], Okto Dinaryanto[2], Mardiana Irawati[3]
[1] Department of Electrical Engineering, Sekolah Tinggi Teknologi Adisutjipto
[2] Department of Mechanical Engineering, Sekolah Tinggi Teknologi Adisutjipto
[3] Department of Informatics Engineering, Sekolah Tinggi Teknologi Adisutjipto
Jln. Janti, Blok R, Kompleks Lanud Adisutjipto, Yogyakarta, Ph. +62-274-451262, Fax. +62-274-451265
*Corresponding author, email: freddykurniawan@stta.ac.id

Abstract

The major weaknesses of traffic controllers in Indonesia are unable to accommodate the variety of traffic volume and unable to be coordinated. To solve the problem, a pre-timed and coordinated traffic controller system is built. The system consists of a master and a local controller. Each controller has a database containing signal-timing plans. To synchronize the signal-timing, the master controller sends the synchronization data to the local controller wirelessly, and the local controller can modify a cycle length by adding or subtracting the green interval of any phases. The transition time for synchronization only takes one to several cycles. The algorithm for controlling the traffic including coordination can be done by an AVR microcontroller. Memory usage of the microcontroller is lower than 10%, meanwhile the CPU utilization is no more than 1%, and thus the systems could be widely developed.

Keywords*: traffic controller, pre-timed, coordinated, AVR microcontroller*

1. Introduction

1.1 Background of the Problem

Traffic jams in most major cities in Indonesia have resulted in losing million rupiahs every hour. Traffic congestion is caused by many factors. One of the factors is the current traffic controller cannot accommodate the variety of traffic volume. Traffic congestion happens usually at the main junctions in the morning, before the office hours (6.00 a.m. to 8.00 a.m.) and in the evening, after the office hours (3.00 p.m. to 5.00 p.m.). Meanwhile, at midnight, sometimes many vehicles have to wait even though there are no others vehicles pass at the intersection. Because the traffic signal remains red for the fixed period, the vehicle should wait until the signal turns to green.

Many research studies have been done to improve the traffic controller in order to accommodate the traffic variation. Fazli obtained vehicles classification base on neural networks for an intelligent traffic controller system [1]. Khan and Askerzade implemented an image processing and fuzzy logic control, then sent the result to a microcontroller to drive the traffic signal in the desired manner [2]-[3]. Hongjin Zhu presented a moving vehicle detection and tracking system, which comprising of horizontal edge detection method and auto correlation. The result shows that it is possible to detect each individual vehicle even if the vehicles are overlapping [4]. Although vehicle detection has been improved, there is no guarantee that the process gives the accurate result. To overcome the problem, the traffic controller should have a signal-timing plan that should be used if there is an invalid result from detection progress. The plan can be defined from traffic volume variation.

Some research studies have been done to improve vehicle flow at the road network. Xie presented a schedule-driven coordination for real-time traffic network [5], and Shamshirband presented an improvement of control ability by using a neural network Q-learning approach as on Game Theory [6]. Lei Wu presented a bee inspired zonal vehicle routing algorithm to provide a reasonable and effective optimal route for the Dynamic Route Guidance System [7]. According to Dotoli, one of the effective methods to improve traffic flow in the network is to synchronize the traffic signals at all intersections especially at adjacent intersections [8]. The objective of the coordinating is to provide continuous flow of traffic along streets or highways.

The *Manual on Uniform Traffic Control Device* (MUTCD) also recommends that traffic signals within 800 meters must be coordinated under a common cycle length. Other reasons to

Received August 18, 2014; Revised October 8, 2014; Accepted October 25, 2014

establish coordination are when the intersections are in close proximity to one another and traffic volumes between them are large. Coordinated traffic controller can permit continuous movement along an arterial or throughout a network of major streets with minimum stops and delays. By minimizing the amount of acceleration and breaking, the air pollutant emitted by the vehicle will be reduced [9]. It can be seen from the previous research that the benefits of coordination depend on the uniformity of traffic flow. However, in the scenario that the flow is not uniform, some improvements can be done in the future [10].

The modern traffic controller system generally is based on 32-bit processor and coordinated by computer systems that act as a server [11]. Meanwhile, present traffic controllers in Indonesia are based on MCS-51 8-bit microcontroller. It is difficult to build the system on the last microcontroller due to the limitation of CPU speed and memory space. The complexity of the algorithm for coordinating traffic controllers could not be done by a MCS-51 microcontroller.

Several research studies have been introduced to meet the traffic characteristic in Indonesia. Primantary presented a coordination of the traffic signal model [12] and Jatmiko presented the architecture of decentralized self-organizing traffic control in real situation even on non-structure intersection like in Jakarta [13]. However, it is found no studies about the coordinated signal concerned with the traffic controller system in Indonesia that is based on microcontroller.

The purpose of the research is to build a pre-timed and coordinated traffic controller system based on an AVR microcontroller. Based on the signal-timing plan, the new algorithm for controlling traffic can accommodate the traffic volume variation and coordinate another traffic controller. The algorithm is tested on an ATmega128A AVR 8-bit microcontroller that has much faster on the CPU and larger memory space.

1.2 Pre-Timed and Coordinated Traffic Controller

A traffic controller is used to switch the traffic signal. The signal sequence is red, green, yellow, and red-clearance. The main variables in signal-timing are cycle length and green split. Cycle length (C) is the time in seconds that it takes a signal to complete one full cycle of indications. One cycle is the sum of the duration of the green interval (g) plus the yellow interval plus the red-clearance interval. It also indicates the time interval between the starting of green for one approach until the next time the green starts. Green split for a signal in a given direction is defined as the fraction of cycle length when the signal is green in that direction.

Pre-timed traffic controllers operate in a predetermined and regularly repeated sequence of signal indications. Generally, this type of traffic controller is cheaper to purchase, and easier to install and maintain than the other types such as actuated traffic controllers. Their repetitive nature also facilitates coordination with adjacent signals, and they are useful where progression is desired [11].

An additional variable of coordination is offset. The offset (θ) can be defined as the interval from an offset reference point at one signal to the following nearest one at the other signal. The offset reference point is a point where the controller makes a decision to terminate the coordinated phase. The cycle length can be measured from the successive offset reference point defined by the operator [11].

2. Research Method

2.1 The System Hardware and Software

The prototype of the system consists of two traffic controllers, one acting as a master controller and the other as a local controller. As used by previous researchers [6],[13], the system uses a distributed systems approach for building a time-based coordination. A traffic controller can be operated for stand-alone or coordinated mode. Each traffic controller has all the features desired for signal control at the intersection such as CPU, accurate clock generator, and communication module as shown in Figure 1. Each traffic controller has a database containing signal-timing plans that will be allocated to manage vehicle flows in the lane for the signalized intersection. Each hour of the week is covered by the signal-timing plan.

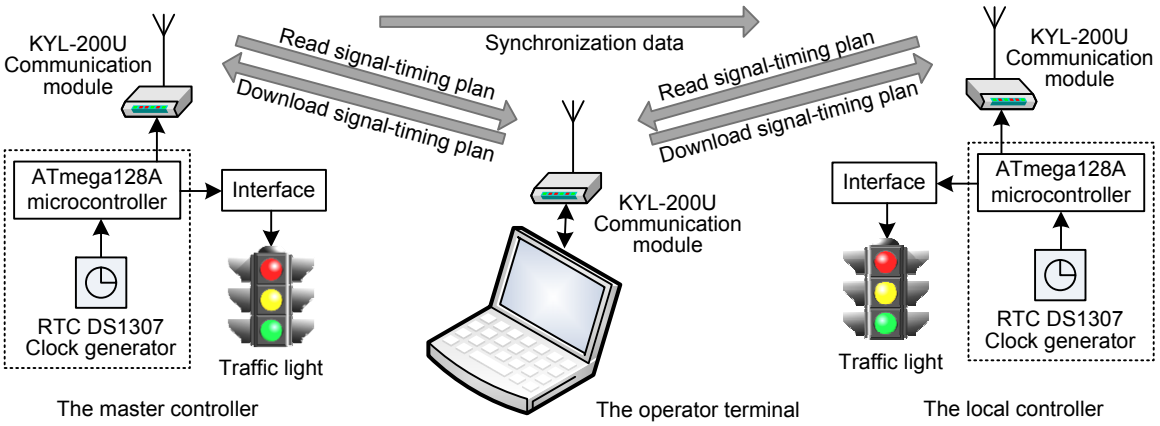

Figure 1. The system block diagram

At the heart of the traffic controller is an ATmega128A AVR microcontroller. The microcontroller is running at a clock frequency of 11.0592MHz. The microcontroller contains main peripherals: an 8-bit RISC microprocessor, 128kB flash memory space, 4kB SRAM, and 4kB EEPROM. The algorithm for controlling traffic is implemented to be a Traffic Controller Program written in C language using CodeVision AVR 2.05.3 Standard Version. The program is stored in the flash memory and the database containing signal-timing plan is stored in the EEPROM of the microcontroller. In order to save the memory resource and reduce the computation load, all variables are formatted as 8 or 16-bit integer and all mathematical computations are performed using integer operations.

A KYL-200U wireless communication module is used for communication at each traffic controller and operator terminal. The operator terminal can download the signal-timing plan to each traffic controller and read the plan from each traffic controller. In order to coordinate the local controller, the master controller also sends synchronization data to the local controller. The module transmits the data using Frequency Shift Keying modulation at baud rate of 9600bps. A frame of the data contains of 8-bit data, a start bit, a stop bit, and no parity. All types of communication are done wirelessly at a frequency of 433MHz with different header data.

2.2 Signal-timing Plan

The research is focused on a couple of a four-way intersection (crossroad) with a traffic signal at each intersection. For each intersection with a set of entry and exit roads, the traffic signal cycles through a fixed sequence of phases. With a signal-timing plan, a day is segmented to ten time slots. To accommodate day-variance of traffic volume in a week, there are three signal-timing plans available that can be allocated to several types of day: weekdays, Saturday, and Sunday. All plans of the master and the local controller are defined at the Traffic Management Center program in the operator terminal and can be downloaded to both controllers wirelessly as shown in Figure 1. All intersections in the area can be controlled by different timing-plans simultaneously that were merged into an overall samaphoric cycle. Once the plans are set, they remain fixed until they are changed manually from the terminal. Table 1 shows the signal-timing plan for weekdays that is used for experiment.

Table 1. A signal-timing plan for the master and the local controller

Parameter	Time slot	Time	Master controller				Local controller					
			P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	θ	p
Green interval (g)	1	4:30 a.m.	10	10	10	10	10	10	10	10	0	0
	2	6:00 a.m.	12	15	20	15	12	14	20	15	0	0
	3	6:30 a.m.	15	20	30	20	15	19	30	20	20	40
	4	7:10 a.m.	17	20	26	20	17	19	26	20	20	20
	5	8:00 a.m.	20	22	26	20	20	20	27	20	5	20
	6	10:00 a.m.	25	25	25	25	21	25	28	25	5	20
	7	3:30 p.m.	30	25	22	25	29	25	22	25	10	20
	8	6:00 p.m.	25	25	20	25	22	22	20	22	10	20
	9	9:30 p.m.	15	15	15	15	0	0	0	0	0	0
	10	12:00 p.m.	0	0	0	0	0	0	0	0	0	0
Yellow interval (Y)	All time		3	3	3	3	3	3	3	3		
Red-clearance interval (R)	All time		5	5	5	5	5	5	6	5		

A time slot of the master controller consists of time to start the time slot and green interval (g) of all phases, while a time slot of the local controller has an additional offset (θ) and limit data (p). There are some timing constraints for safety and fairness at phase i (P_i) : the green split (g_i) runs for a variable interval that can range between a minimum (G_{min}) and maximum (G_{max}) while the yellow and red-clearance splits run for a fixed interval (Y_i and R_i). If the green interval of all phases at a time slot is zero, the controller will not control the traffic at the time slot and the signal displays flashing-yellow. Offset can vary from zero to the cycle length. All values have been rounded into numbers of time step of one-second resolution.

As mentioned previously, in order to synchronize the local controller, the master controller sends the synchronization data (sync). When receiving the data, the local controller reads the time from the RTC then saves the time as t_M. Afterward, at the end of local cycle (t_L), the microcontroller of the local controller executes algorithm 1 for synchronizing the signal-timing of the local controller to the signal-timing of the master controller.

Algorithm 1: Synchronize the signal-timing of the local controller to the master

□□ = read_time(RTC)

$$\square = \sum_{\square=1}^{4} \square_{\square}$$

$$\square_{\square\square\square\square\square} = \square.\square$$

$$\square' = \square_{\square} - \square_{\square}$$

$$\xi = \square' - \theta$$

if ξ = 0 or ξ > □ then return

if ξ > 0 then go to 13

if ξ > □□□□□□, then ξ = □□□□□□

$$\Delta\square_{\square} = \frac{\square_{\square}}{\sum_{\square=1}^{4} \square_{\square}} \times \xi$$

$$\xi = \xi - \sum_{\square=1}^{4} \Delta\square_{\square}$$

$\xi \rightarrow \Delta\square_{\square}$

for i =1 to 4 do $\square_{\square} = \square_{\square} + \Delta\square_{\square}$

return

if ξ < – □□□□□□, then □ = – □□□□□□

$$\Delta\square_{\square} = \frac{\square_{\square}}{\sum_{\square=1}^{4} \square_{\square}} \times \xi$$

$$\xi = \xi - \sum_{\square=1}^{4} \Delta\square_{\square}$$

$\xi \rightarrow \Delta\square_{\square}$

for i =1 to 4 do $\square_{\square} = \square_{\square} - \Delta\square_{\square}$

return

The first step of the algorithm (line 1) is to read the time from the RTC and store it as 16-bit integer t_L. After that, line 2 computes a total of the green interval of all phases (*G*). Line 3 computes the G_{limit} that will be used to limit the maximum amount of adjustment that can be made in one cycle. The parameter is used to prevent an excessive green interval or green intervals that are too short during transitioning. If a limit is imposed, the signal may not be able to complete offset transition within one cycle. The algorithm constrains to add or subtract the green interval no more than a certain percentage of p that ranges from 0 to 99. The value of 0 means the local controller is not synchronized to the master controller.

Line 4 computes the current offset that is defined as the time difference between the end of the master cycle (t_M) and the end of the local cycle (t_L). The value is stored in a 16-bit unsigned integer (θ'). If the current offset is equal to the offset defined in the database (θ), the signal-timing of the local controller has been synchronized to the signal-timing of the master controller. A synchronized local timing can be shown as case 1 in

Figure 2.

Let ξ denotes the time difference between θ' and θ in line 5. If ξ is less than zero, t_L is leading, then lines 8 – 12 shift next t_L progressively later by timing the next cycle length slightly longer than the programmed cycle length. Synchronizing a leading local timing can be shown as case 2 in

Figure 2. On the other hand, if ξ is greater than zero, t_L is lagging, then lines 14 – 18 shift next t_L progressively earlier by timing the next cycle length slightly shorter. The case can be shown as case 3 in

Figure 2.

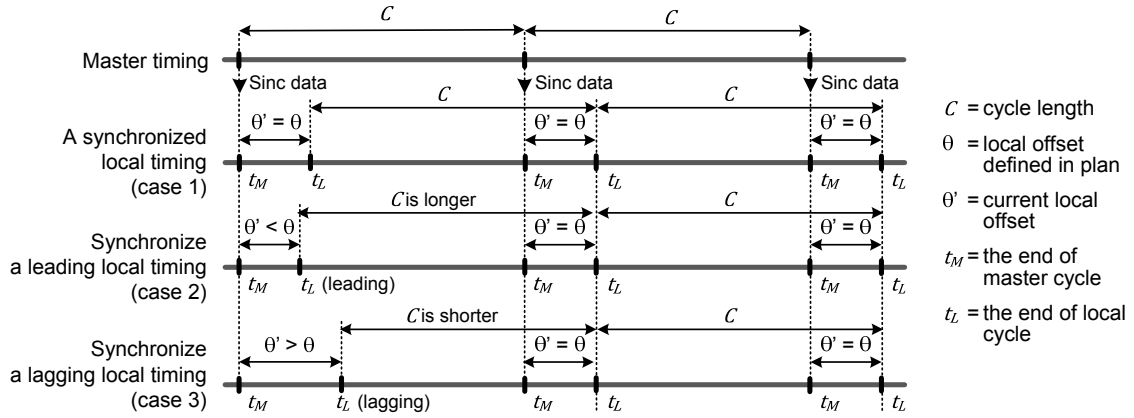

Figure 2. Shifting the end of local cycle

When the next t_L will be shifted progressively later, line 9 distributes extra green interval proportionally amongst all phases. The higher the green interval of a phase (g) thus the higher the addition in the green interval (Δg_i). The equation in line 9 is done by multiplying g_i and ξ first, then the corresponding result in an 16 bit integer is divided by $\sum_{i=1}^{4} g_i$. Note that the computation is performed using integer division. An 8-bit integer part of the division result is a change in the value of the green interval (Δh_i); meanwhile the remainder is distributed to the addition of any green interval starting from the phase that has the greatest value of green interval. Finally, line 12 computes the green interval that will be allocated to the next cycle.

Lines 14 – 18 are to shift t_L progressively earlier by subtracting time from one or more green interval in the sequence in the similar manner. When this method is used, the new green interval may be close to the minimum phase green interval (G_{min}), meaning only a small adjustment in cycle length can be made by shortening. In the case, it may take many cycles to complete an offset transition.

3. Results and Discussion

The system is tested using a signal-timing plan for the master and the local controller as shown in Table 1. The master controller controls the vehicle flow at time slot 1 until 9 (4.30 a.m. ~ 11.59 p.m.) and the local controller controls at time slot 1 until 8 (4.30 a.m. ~ 9.29 p.m.). Time slot 1 and 2 of the local controller are not coordinated to the master controller because $\theta = 0$ and $p = 0$.

3.1 Transition Time

In order to measure the performance, the system is tested by operating it on difference value of offset error (ξ). It operates using time slot 5 that the corresponding time is from 8.00 a.m. to 9.59 a.m. Both controllers call a 120-second cycle length while the local controller also calls for a 40-second offset. The master controller restarts at 08:00:00 and the local controller restarts at 08:00:40. In the case, the current local offset is equal to the offset defined at the time slot. A dashed-gray line in

Figure *3* shows the diagram of the cycle length of the case. There is no transition time because there is no offset error value occurs in the transition (ξ = 0s).

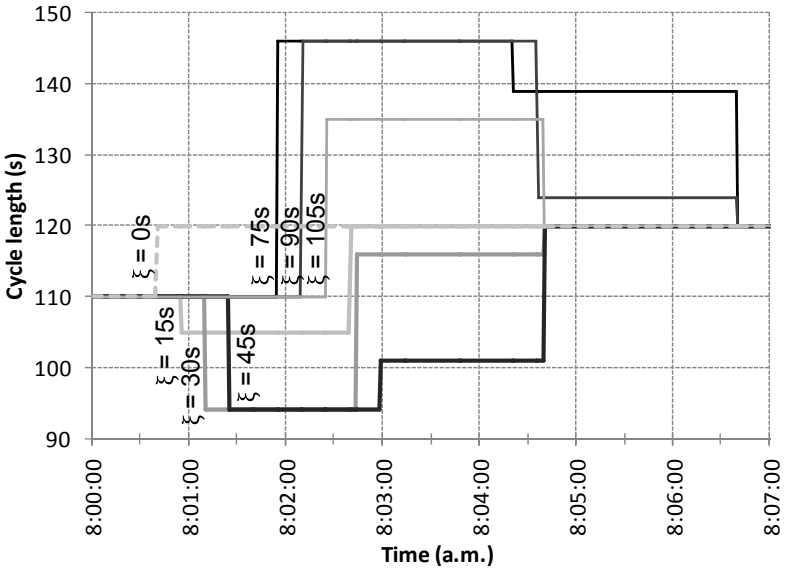

Figure 3. The fluctuation of cycle length versus different of ξ

In the second case, the local controller restarts at 08:00:55 thus it has a 55-second local offset. The offset error value (ξ) is 15 seconds. In order to synchronize the timing of the local controller, the following cycle is subtracted by 15 seconds to be 105 seconds. The transition time is 105 seconds.

In the next cases, the local controller restarts latter thus the current local offset respectively, are 30, 45, 75, 90, and 105 seconds. It can be shown in

Figure *3* that the diagram of the cycle length at the transition time vary with the offset error ξ. When the offset error ξ is less or equal to a half of a cycle length, the next cycle length is decreased in order to shift the next end-of-a-cycle earlier. On the other hand, when the offset error ξ is greater than a half of the cycle length, the next cycle length is increased in order to shift the next end-of-a-cycle later.

Figure *4* shows the comparison of the transition time will be taken versus the current offset in the difference of *p*. The value of *p* will limit the amount of the change in the total green interval. When the system uses the *p* value of 20%, G_{limit} is 16 seconds (computed on line 3 of algorithm 1). The amount of addition or subtraction of the cycle length (ξ) will be no more than 16 seconds. It can be shown in the

Figure *4* (case p = 20%), when the offset error (ξ) is greater than 16 seconds, the transition will take more than one cycle. In general, when ξ is equal or less than a half of the cycle length, the larger the value of ξ, the longer the transition will take. In the case, the cycle will be slightly shorter during the transition. Otherwise, when ξ is greater than a half of the cycle length, the larger the value of ξ, the shorter the transition will take. In the case, the next cycles will be slightly longer during the transition.

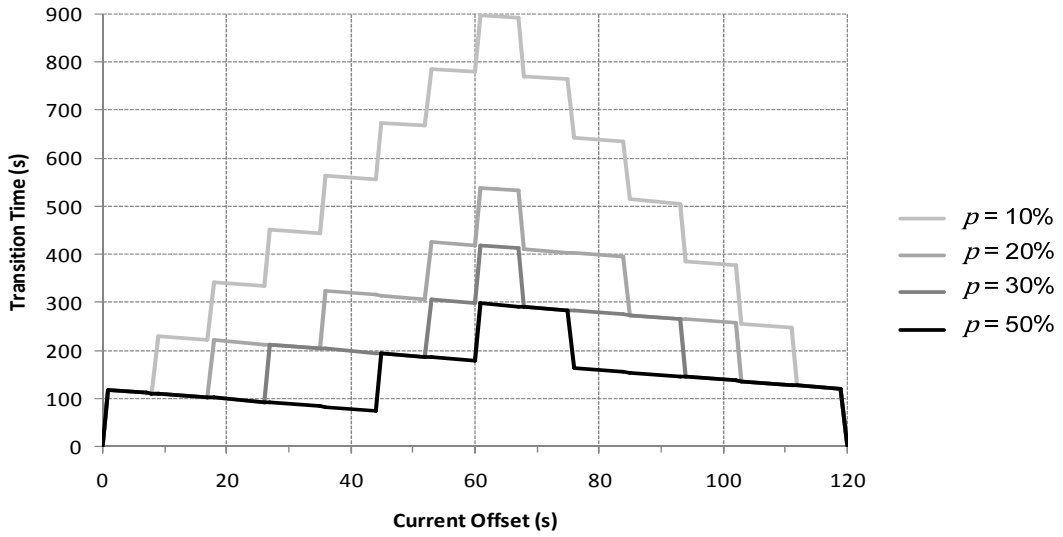

Figure 4. The transition time versus offset

Figure *4* also shows that the larger the value of p, the shorter the transition will take. When the system constrained to add or subtract the total green intervals no more than 10%, a 900-seconds is required to achieve timing synchronization when the value of ξ is from 60 seconds to 68 seconds. However, a lower value of p will make the transition smoother.

3.2 Memory Usage and CPU Utilization

The Traffic Controller Program at the master controller uses 6.5% of flash memory space and 6.2% of SRAM space only as shown in Table 2. The memory usage of the local controller is greater than the master because there is a coordination algorithm at the local controller. Meanwhile, the database contains the signal-timing plans of master controller takes place 4.7% of EEPROM space only and the database at the local controller takes place a larger space due to the addition data for offset (θ) and limiter (p). The report of memory usage can be shown in Table 2.

Table 2. Memory usage of the microcontroller

Memory types	Capacity (Byte)	Memory used			
		Master		Local	
		(Byte)	(%)	(Byte)	(%)
Flash	131072	8574	6.5	10914	8.3
SRAM	4351	271	6.2	337	7.7
EEPROM	4096	192	4.7	252	6.2

In general, when executing the Traffic Controller Program, in the period of one-second interval available, the CPU only spends less than 10 milliseconds on processing the Traffic Controller Program. The CPU utilization is less than 1%; this means that in most of the time the

CPU in idle state. From the fact that memory usage and CPU utilization are low, it can be concluded that Traffic Controller Program can be expanded widely by adding any features of the traffic controller system without changing the microcontroller.

4. Conclusion

A pre-timed and coordinated traffic controller system can be implemented based on an ATmega128A microcontroller. The system uses the signal-timing plans for daily and weekly basis in a database that will be allocated to manage vehicle flow. The timing of the local controller can be synchronized to the timing of the master controller. The transition time for synchronization only takes one to several cycles. The memory usage of the microcontroller is lower than 10%, while the CPU utilization is no more than 1%, thus the system can be widely developed.

Acknowledgement

The research is fully supported by The Ministry of Education and Culture Indonesia under grant "*Hibah Bersaing Dikti*". The authors would like to thank to "*Kopertis Wilayah V*" and "*STT Adisutjipto*" for their support.

References

[1] Fazli S., Mohammadia S., Rahmani M. Neural Network based Vehicle Classification for Intelligent Traffic Control. *International Journal of Software Engineering & Applications*. 2012; 3(3): 17-22.
[2] Khan BA, Lai NS. An Advanced Fuzzy Logic based Traffic Controller. *International Journal of Innovation in the Digital Economy*. 2014; 5(1): 31-40.
[3] Askerzade IN, Mahmood M. Control the Extension Time of Traffic Light in Single Junction by Using Fuzzy Logic. *International Journal of Electrical & Computer Sciences IJECS – IJENS*. 2010; 10(2): 48-55
[4] Zhu H, Fan H, Guo S. Moving Vehicle Detection and Tracking in Traffic Images based on Horizontal Edges. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(11): 6477-6483.
[5] Xie XF, Smith S, Barlow G. Schedule-Driven Coordination for Real-Time Traffic Network Control. *Proceedings 22nd International Conference on Automated Planning and Scheduling* (ICAPS). Atibaia, Sao Paulo, Brazil. 2012: 323-331.
[6] Shamshirband SS. A Distributed Approach for Coordination between Traffic Lights based on Game Theory. *The International Arab Journal of Information Technology*. 2012; 9(2): 148-153.
[7] Wu L, Yang L, Liu H, Zhang Y. Bee Inspired Zonal Vehicle Routing Algorithm in Urban Traffic. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(9): 6699-6710.
[8] Dotoli M, Fanti MP, Meloni C. Coordination and Real Time Optimization of Signal-timing plans for Urban Traffic Control. *IEEE International Conference on Networking, Sensing and Control*. 2004; 2: 1069-1074.
[9] Dobre C. Using Intelligent Traffic signals to Reduce Vehicle Emissions. *International Journal of Innovative Computing, Information and Control*. 2012; 8(9): 6283-6302.
[10] Wang K, Huang L, Zhai Y, Wang Y. Traffic Signal Synchronization in Double Rings Road Structure. *Journal of Information & Computational Science*. 2012; 9(2): 395-402.
[11] Federal Highway Administration. *Traffic Signal-timing Manual*. Federal Highway Administration. U.S. Department of Transportation. 2008; 6·1-6·49.
[12] Primantari LFA. Koordinasi Pengaturan Lampu Lalu Lintas (Studi kasus: Ruas Jalan Prof. Dr. Sorharso - Adi Sucipto - A.Yani - Adi Soemarmo). *Jurnal Teknik Sipil dan Arsitektur*. 2010; 8(12): 13-30.
[13] Jatmiko W, Azurat A, Wibowo AH, Marihot H, Wicaksana M, Takagawa I, Sekiyama K, Fukuda T. Self-Organizing Urban Traffic Control Architecture with Swarm-Self Organizing Map in Jakarta: Signal Control System and Simulator, *International Journal on Smart Sensing and Intelligent Systems*. 2010; 3(3): 443-465.