U.P.B. Sci. Bull., Series D, Vol. 84, Iss. 1, 2022 ISSN 1454-2358

DESIGN OF TRAFFIC INTERSECTION MODEL AND REGULATION WITH SOFTWARE AND MICROCONTROLLERS

Ilir DOÇI[1], Ramadan DURAKU[2*], Bashkim HOTI[3]

This paper describes the development of a 3D printed model of traffic intersection with traffic lights, implementation of detectors, programming, and regulation using the Software and Microcontrollers. The model is developed and regulated in a coordinated manner as a system with hardware, electronic components, and programming using the software. It is a proportionally small model of the cross-typed intersection with two lanes in each direction and four traffic lights. In the model are also implemented Infrared sensors for the detection of vehicles in a queue and the length of the queue. Synchronization of Traffic signals and regulation of detectors are programmed with LabVIEW Software and controlled by an Arduino MEGA microcontroller. The aim is to optimize the travel phases of vehicles in the intersection to collect data and regulate them in real-time using LabView software. It evaluates the traffic parameters and optimizes the traffic signals, to reduce traffic delays and queues.

Keywords: Regulation, Traffic intersection, 3D modeling, traffic lights, programming, LabView, microcontroller.

1. Introduction

Modern Urban traffic Intersections have problems with the Regulation and Control due to high traffic volume and demand that come because of the high number of vehicles in traffic. This paper presents a methodology of regulation and control of vehicle's motion and signalization in the Cross-type Traffic Intersections. This is done with the implementation of 3D physical model design (prototype), programming and simulations using *LabView Software* [1], [2], implementation of *Arduino Mega microcontrollers* for intelligent systems [3], [4] and selection of Infrared sensors as detectors for vehicles detection and queue control [5].

[1] Prof., Dept.of Mechatronics, University of Prishtina, Faculty of Mechanical Engineering, Kosovo, e-mail: ilir.doci@uni-pr.edu

[2*] Prof., Dept.of Traffic & Transport, University of Prishtina, Faculty of Mechanical Engineering, Kosovo, e-mail: ramadan.duraku@uni-pr.edu. (Corresponding Author*).*

[3] Student, University of Prishtina, Faculty of Mechanical Engineering, Kosovo, e-mail: bashkim.hoti.98@hotmail.com

Several authors have published works related to the design of traffic intersection models, including the implementation of software for the control and programming.

In the papers on the topic of Traffic Model Development, authors Slinn et al. [6], Doçi [7], De Sousa [8], Salter R.J [9], have presented design principles and theory of model development for traffic systems. Authors Barcelo [10], A.D. Ambrogio et.al [11] have presented the principles of simulation of traffic intersections. Lebacue [12], Fabianova et al. [13], Erdmann et al. [14], Doçi et al. [15], [16], Krasniqi et al. [17] have analyzed models of traffic intersections using various traffic applications.

Regarding the development of Traffic models using LabView software, Temani et al. [18] have presented the approach of managing traffic systems using Lab View software. Kumar et al. [19] and Biradar et al. [20] have presented implementation of traffic control system using LabView programming with model prototypes. Patrascoiu et al. [21] have given the method of the implementation of a traffic signal controller with LabView Programming. Srinivas et al. [22] have worked on the traffic control system with LabView. Dhiman et al. [23] have presented the method of modeling and regulation of Traffic Light using LabView. Doçi and Duraku in their teaching book [24] have given several examples of traffic models regulation using Lab View software.

Authors M. Keanu et al. [25] and N. Shahrubudina et al. [26] have shown materials for 3D printing and types of 3D printing applications and technologies, used in printing technology. Materials and filaments mentioned in their works are used to develop the model in this paper.

Concerning the LabView programming and control, Richard Jenning et al. [27] presents the LabView Graphical programming fundamentals implemented in this paper. Manuals of LabView are published from National Instruments Corporation concerning the main and advanced features of the LabView software [1], [2], and Data acquisition from the devices [28], [29].

Sources regarding the connection and communication between Arduino Mega microcontrollers and LabView are referenced in [30] and [31].

Cross-typed intersections with traffic lights are typical examples of traffic nodes that require optimal regulation of vehicle motions and other participants in traffic, proper timing of travel phases, control of queues in case of high volume and traffic jams [7]. It is important to design them properly and implement regulation and control for optimal flow of traffic [6], [10]. For the study purpose and implementation of Methodology, a proportionally small physical model of Cross-type Intersection is designed using Light Materials and 3D printing (Fig.1).

Fig.1. Model of Cross-type Intersection with Software Control

The Intersections of Type "+" operate according to the phase adjustment and the priority within the cycle based on the traffic loads according to the directions (E-W or N-S). The Main Regulation task in the model is programming the priority of vehicles passing on Through Lanes by turning Green Light in optimal timing [10]. This includes assigning priority for those Through Lanes that have queues of vehicles waiting to pass through Intersection. Through Lanes are East Bound Through (EBT), West Bound Through (WBT), South Bound Through (SBT) and North Bound Through (EBT) [16], [32] (Fig.2).

Based on the model created, shown in Fig.1 & Fig. 2, there are two travel phases of vehicles that will be regulated in the Lanes of Intersection: Left – Right direction (EBT & WBT) and North-South direction (NBT and SBT) [15]. Dimensions of the Intersection model are 80x80 cm. There are 4 traffic lights implemented in the model with three color signals: Red, Yellow, Green. There are 8 Infrared sensors as detectors, two for each Through Lane [5]. They are positioned in two places in Lane Length: Close Proximity Detector (CPD) is at 1 cm distance from the Traffic Light and Distant Proximity Detector (DPD) is at 25 cm distance from the Traffic Light (Fig.2). The task of CPD Detector is to detect if there are vehicles waiting to pass the Intersection and the task of DPD Detector is to detect if there are long queues in the Through Lane. The aim is to create an algorithm and programming to assign priority at any Through Lane that has vehicles waiting and queues of vehicles, and to turn the Green Light on the Traffic Light in the shortest timing. At the same time, the Traffic Lights in other through lanes will get the command to turn Red Light.

The Regulation process also gives priority to Lanes of the same direction, if any of them fulfills the criteria ordered by detectors.

For testing and verification purposes, there were purchased 15 small models of metallic vehicles of various types: Passenger vehicles Models, Bus and Truck models.

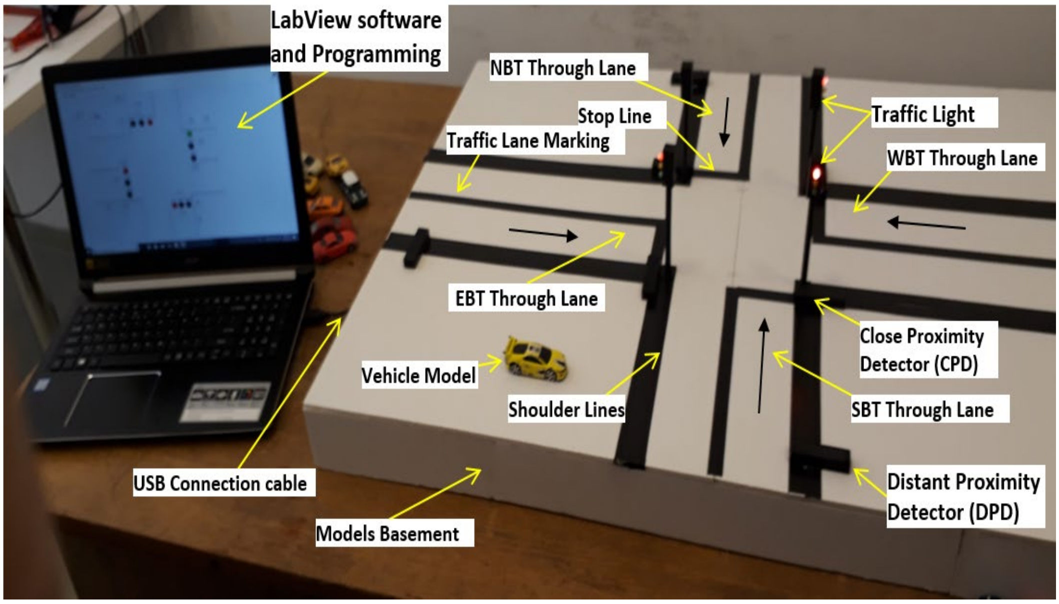

Fig.2. Main parts of the Model of the Intersection

2. Design of the Intersection

The invention of 3D printing provides new alternatives for the development of models of Traffic Intersections and other traffic Systems. Traffic Lights with their parts and Detector Covers are 3D printed with moulded plastic materials.

Plastic materials as filaments are ABS – *Acrylonitrile Butadiene Styrene* and PBS – *Polybutylene succinate* [36]. Parts are developed one by one using 3D printer [26]. The basement of Intersection is designed using *EPS- Polystyrene Foam*. Traffic Lights are 12 cm Long and have mounted 5 mm LED Diodes of 5V for red, yellow, and green colour signals.

This phase consists of planning the Intersection structure, materials selections, design through 3D printing, mounting, and joining of parts. In Fig.3 is the 3D printer used, which is developed at the Laboratory of Mechatronics of the Faculty of Mechanical Engineering [33].

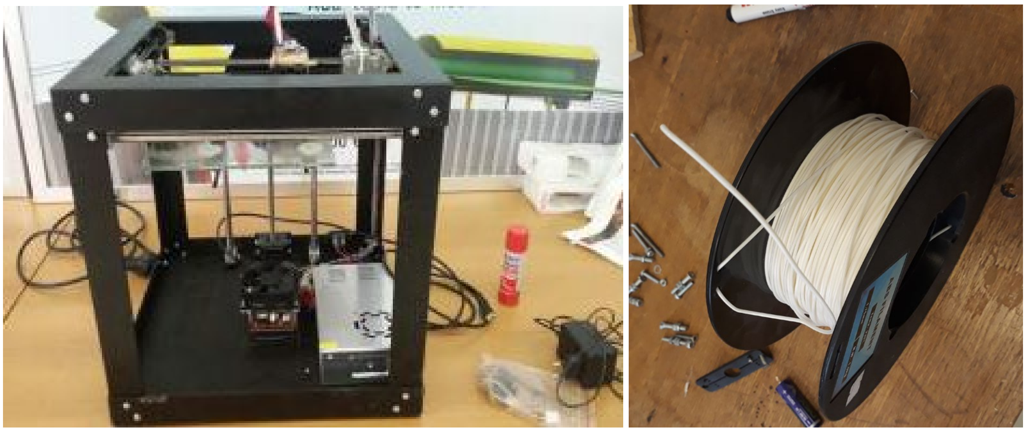

Fig.3. 3D printer developed at the Laboratory of Mechatronics, and the filaments used

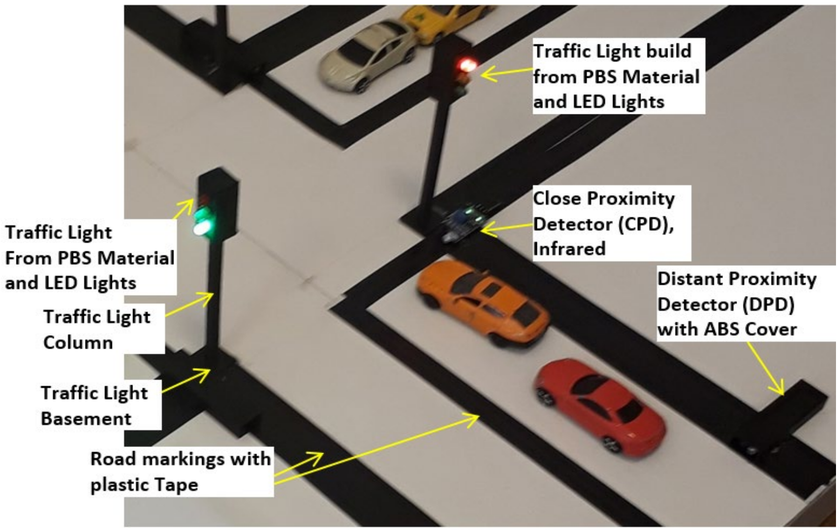

Fig. 4. Main Parts created with 3D printing

3. Implementation of Hardware for Regulation and Control

After the development of Intersection and mounting of parts, the next phase consists of wiring, regulation of signalization and implementation of control. Main control hardware is mounted below the Intersection Basement, including cable links with Computer (Fig.5).

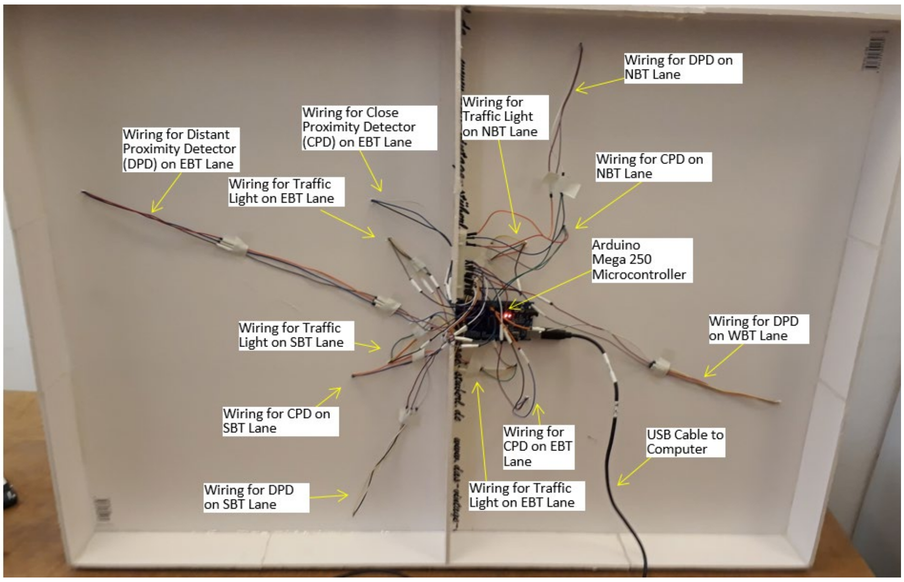

Fig.5. Main Hardware Parts and wiring mounted below the Intersection Model

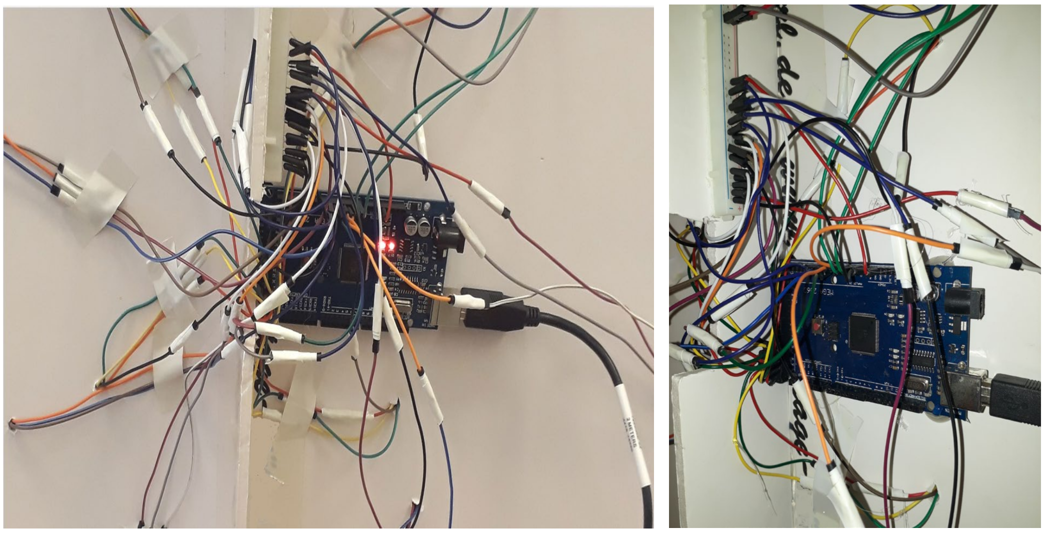

Fig.6. Connections and wiring on Arduino Mega 2560 Microcontroller

3.1. Arduino Mega 2560 Microcontroller implementation

Arduino Mega is an open-source and single-source microcontroller that can be programmed to analyze and produce electrical signals to perform a variety of automated tasks such as sensor control, lighting control, regulation of robotic equipment, actuator control, etc. (Fig.6) [3]. It is a platform based on the I/O interface and programmed with the Wiring processing language like C ++. In this model, Arduino Mega 2560 will be used to connect with the computer and communicate with Lab View software, while signals from Arduino Mega will be transferred to 4 Traffic Lights and 8 InfraRed detectors for the Regulation of vehicle's travel in the Intersection [5]. In Fig.7 are given Arduino Mega Pinouts with explanations [4]. The power supply is achieved by connecting it to a PC using a USB cable, battery or an AC-DC adapter. There are 54 digital Input/Output Pins (of which 14 can be used as PWM outputs) 16 Analog pins, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The digital pins are used to interface sensors by using them as input pins or drive loads by using them as output pins. The operating voltage is 5 V. The recommended Input Voltage will range from 7 V to 12 V.

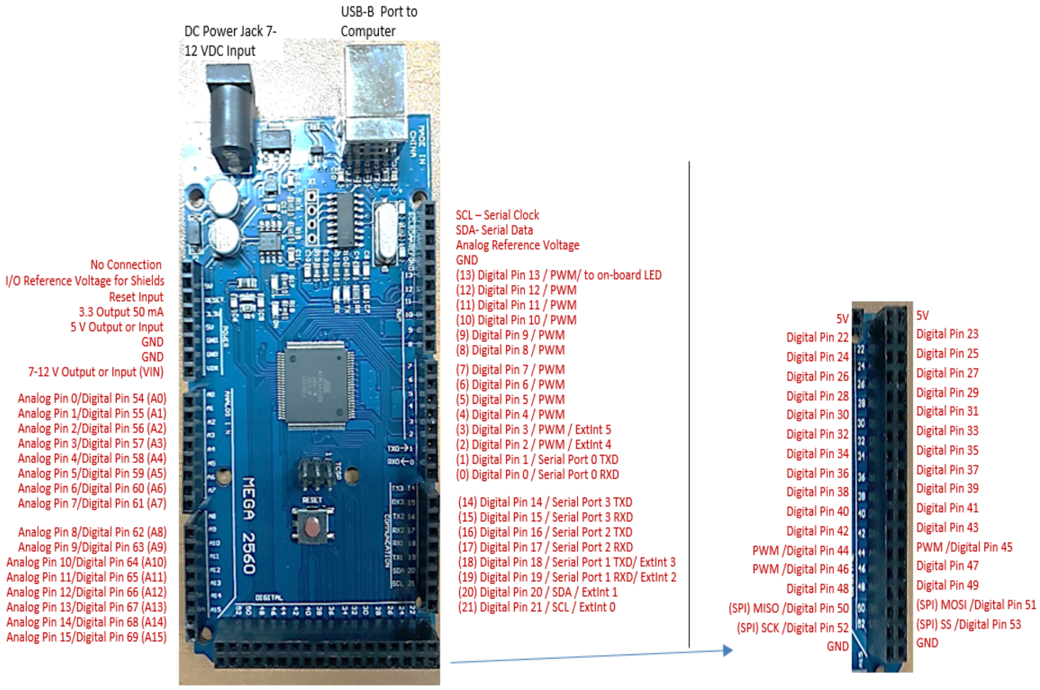

Fig.7. Arduino Mega 2560 Microcontroller with Pinouts [19]

3.2. InfraRed Sensors (Detectors)

IR (Infra Red) sensors will be used to detect vehicles in the intersection. Vehicles will be detected based on the IR light transmitter and receiver, and then, according to the vehicle presence in the Lane and queues created the traffic lights will be autonomously regulated. The type of Sensor *IR Infrared Obstacle Avoidance Sensor Module for Arduino* [5]. The sensor module is adaptable to ambient light, having a pair of infrared emitting and receiving tubes, transmitting tubes emit an infrared certain frequency, when the direction of an obstacle is detected (reflection surface), the infrared reflected is received by the reception tube (Fig. 8).

The sensors are positioned in separate positions on each Lane, two for each Lane. Based on the specified interval within each sensor that detects vehicles, it will provide signals and values by which road priorities are set. Traffic signals are automated based on the signals from sensors and values calculated by the Arduino Microcontroller, and from this we can monitor and control the traffic density present in the particular area of the Intersection.

The principle of sensor work is as follows: when there are no objects in front of the sensor, the OUT gate output level is 5 V continuously. When the module detects obstacles ahead of the signal, at the OUT port we will have a constant signal level of 0 V. The Sensor's detection range is 2-30 cm, and detection angle 35°. The distance can be adjusted by the potentiometer, if we rotate it clockwise the detection distance increases, and counterclockwise decreases the detection distance.

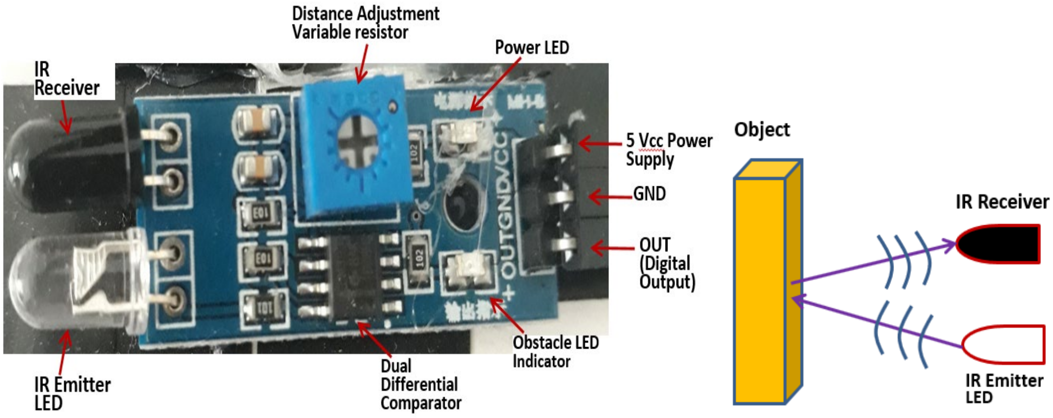

Fig.8. Sensor type IR Infrared Obstacle Avoidance Sensor Module for Arduino and principle of work

4. Control Development with LabView Software and Programming

Programming for the regulation of detectors (InfraRed sensors) and Traffic Lights signaling in the Intersection is achieved using LabView Software [1]. This software is chosen because it offers many possibilities of modelling, visualization, calculation, simulation, and control. The software creates programs using Data Flow programming with wired diagrams.

In Figure 9 is presented the General Algorithm of Regulation and Control of Intersection.

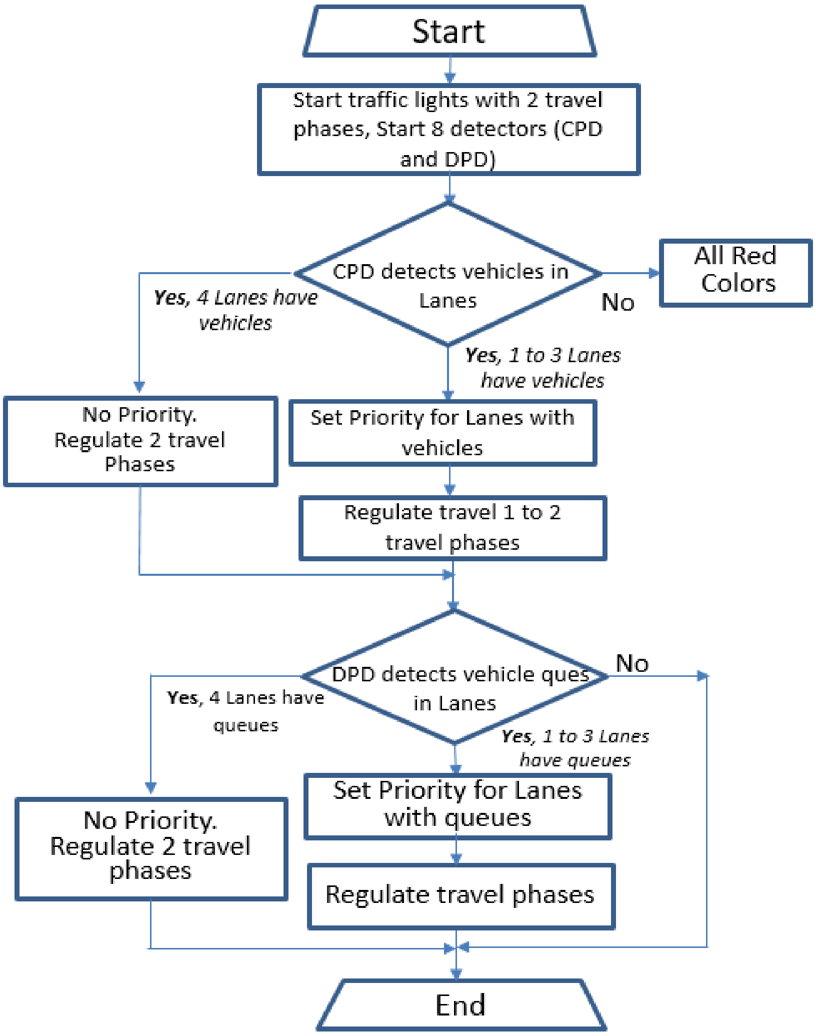

Fig 9. General Algorithm for the regulation of the Intersection

Based on the Algorithm in Fig. 9, the next task is the development of the Program in LabView software. It is shown in Fig.10 & Figure 11. This software has two panels for the development of the program: *Front Panel* and *Block Diagram*. These two panels are interconnected, meaning the functional elements created in Front Panel will be present and functional also in Block Diagram [2], [1], [24].

In the Front Panel, in Figure 10, is designed the Control Board of Intersection with important elements and variables. In this panel are created necessary Input/Output variables and other control parameters of Traffic Lights and IR Sensors as Detectors. The program will run in Continuous mode.

On the Front Panel, in real-time we can track the status of each Traffic Light as well as the entry-exits of the vehicles detected by sensors on each Lane, and the actual number of vehicles on those Lanes.

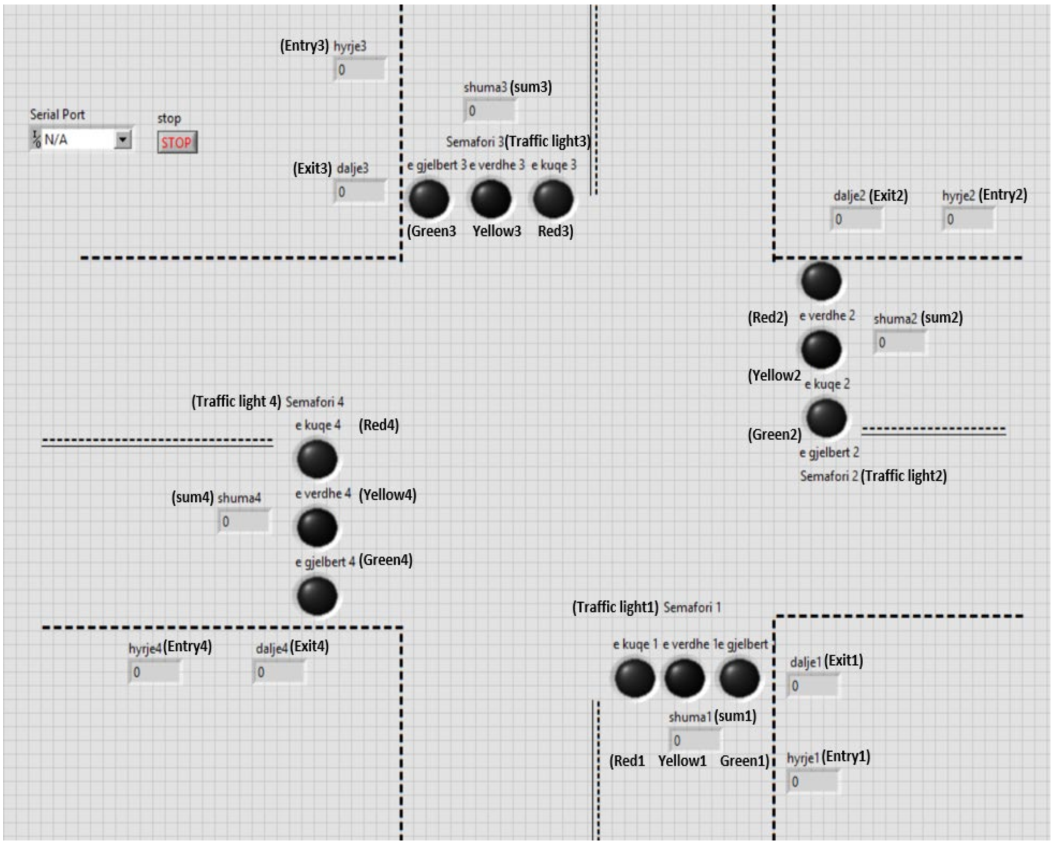

Fig.10. Program designed in the Front Panel

In the Block Diagram, in Fig. 11, are given programming procedures, codes, and connections with virtual wiring elements that are developed in the Front Panel. Here are added other elements necessary to make the program functional like While Loop structure, variables, constants, select operators, LINX elements, etc [37]. In this phase, the program is tested and further improved.

Receiving signals from sensors and transmitting signals for the traffic lights – Based on the programming code shown in Fig. 11, the 8 analog Arduino ports are used for the 8 sensors. In each Lane, two sensors are activated as input (Entry *n*) and output (Exit *n*) of vehicles detected in the particular Lane and made accurate calculations of the number of vehicles found (*n* is the number of Lane, 1 – SBT, 2-WBT, 3-NBT, 4-EBT).

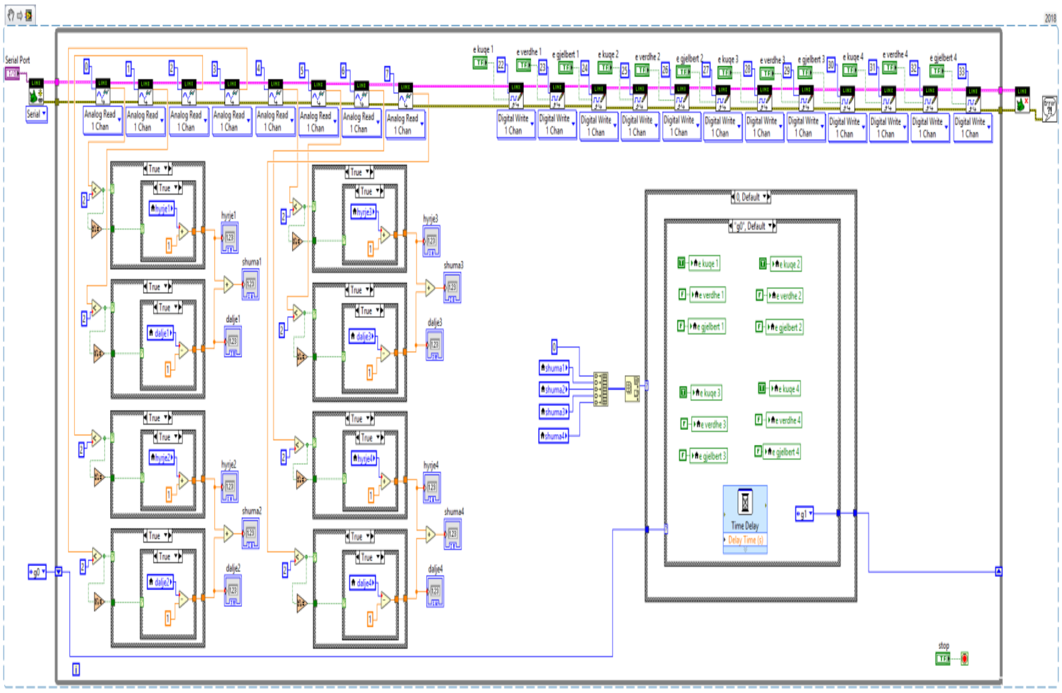

Fig.11. Program in the Block Diagram

The12 digital Arduino pins are used to activate the traffic lights, three pins for each traffic light to change color signals (Red *n*, Yellow *n*, Green *n*). At first, the number of vehicles on each Lane is calculated and compared, and then based on that calculation is executed the selection of the new condition of signaling in the traffic lights (sum *n*). Default time delays for the travel of vehicles with *No priority* (Fig. 9) are set as follows: This procedure is repeated continuously.

In Figure 12 is shown LINX Maker Hub interface software that is used as connector and converter between LabView software and Arduino Mega2560 Microcontroller [28]. In the Drop-Down Box named *Device Family* is Chosen Arduino Family of microcontrollers, in the Drop-Down Box named *Device* Type, is chosen Arduino Mega2560. Upload Method is through USB Port.

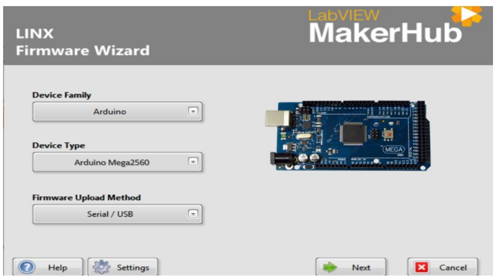

Fig.12. LINX Maker Hub interface software [34]

5. Case Studies of Regulation and Control

In the following Figures are given typical cases of regulation and control for the vehicles passing through Intersection.

Case 1 - In Figure 13 is presented the case when all the Traffic Lights are in Red Colors. This is the case when Distant Proximity Detectors (DPDs) detect no vehicles approaching the intersection from any Line. The sensors will not activate (Change Colors) in any of the Traffic Lights until a vehicle approach towards the particular Traffic Light. This form of the regulation is applied in various Smart Traffic Intersections, in order to allow the pedestrians to pass the Intersection [6], [7]. There are no Traffic parameters to calculate.

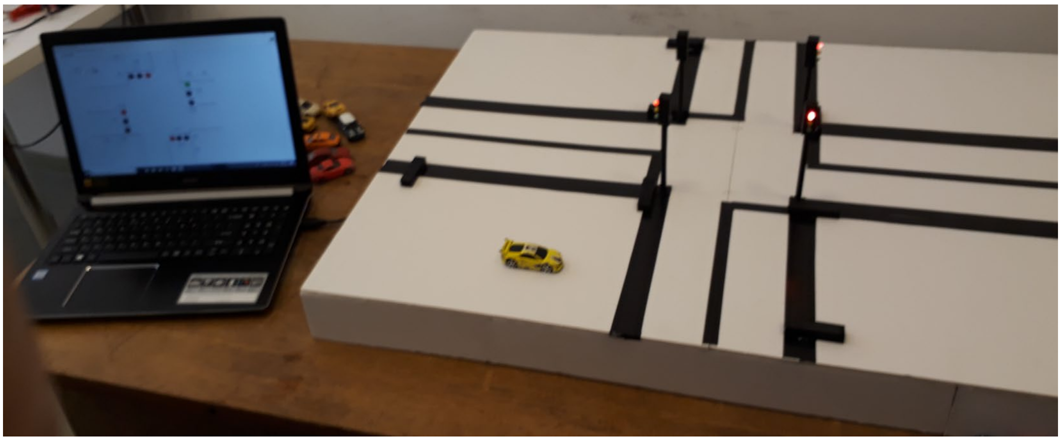

Fig.13. Case 1. No vehicles in Lanes. All Red Colors in Traffic Lights

Case 2 – In the case when one vehicle is coming from outside of Intersection towards a Detector in a Lane, we have the activation of Detectors. This is the second Case of Study, shown in Fig.14. The vehicle is approaching from the Direction EBT, and at the moment when it passes the DPD sensor, this sensor will be activated. The Program will wait until the vehicle reaches the Close Proximity Detector (CPD), and then it will activate the Yellow Color (4 s Length) and then Green Color in the Traffic Light. The vehicle will halt in the Stop Marking of the Intersection, until The Green Light appears, after the Yellow Light. The Distance between Detectors is $L_{ebt} = 25\ cm$, to fit 4 small vehicle models, type passenger cars. At real Intersections, this distance should be at least 20 to 25 m, to fit four passenger cars as a converted unit [10], [32].

While no other vehicles are approaching the Intersection, the Green Color in this Traffic Light will remain until the vehicle passes the Detector, and will remain so for 6 s, until the vehicle passes the Intersection. It is assumed that time

6 s is enough for a vehicle to pass the Intersection [32]. Then the Traffic Light will turn Red.

In total for the Intersection, the Cycle Length is $C_{TOT} = \Sigma(G_i+t_{Li}) = G_{1min}+t_{L1} = 6 + 4 =$ **10 (s).**

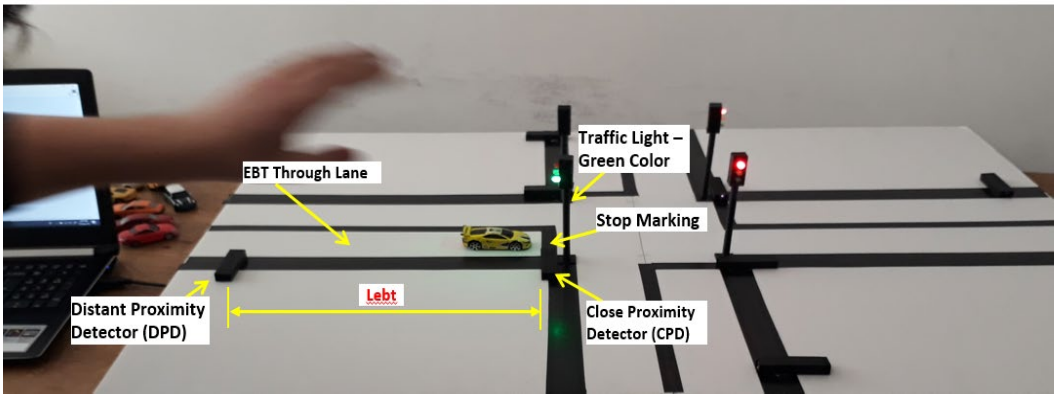

Fig.14. Case 2. One vehicle enters EBT Lane. Receives priority with Green Color

Case 3 - In this case, all the Lanes have one vehicle waiting for the Traffic Light (Fig.15). Based on the traffic rules, there is no priority in the transfer. Therefore, the rule is that two traffic phases are created:

- Phase 1 – EBT – WBT
- Phase 2 – NBT – SBT.

A phase means the Green Light will turn in Two Traffic Lights in two opposite Lane directions of vehicles through motion.

In all four directions, there are vehicles present and detected by CPD detectors. In this situation is applied the priority, according to the regulation of the two-phase plan (EBT-WBT) and (NBT-SBT). But it can also be the opposite, if in the direction (NBT-SBT) there are vehicles that are detected by the CPD detector then the priority is phase 2 (NBT-SBT) and after that the other phase (EBT-WBT). When Phase 1 is Green, then the Phase 2 will remain Red.

Based on calculations, for this Case, the entire Cycle length will be:

Phase 1 - EBT - WBT Lane, total for the Phase is $C_{L1} = 6$ s + 4 s = 10 s,
Phase 2 - EBT - WBT Lane, total for the Phase is $C_{L2} = 6$ s + 4 s = 10 s,

In total for the Intersection, the Cycle Length for this Case is $C_{TOT} = \Sigma(G_i+t_{Li}) = 10 + 10 =$ **20 s**.

We did not implement the Intermediate *All Red timing*, which is found in some Intersections, because it is a variable in timing [10], [32].

Fig.15. Case 4. All Lanes have vehicles. No priority. Yield on 2 traffic phases

Case 4 - In this case, all the Lanes have vehicles waiting for the Traffic Lights (Fig.16). However, not all the Lanes have the same number of vehicles waiting in a queue. In this case, the priority will be based on the number of vehicles in a Lane so that we don't have long waiting times, meaning the Lanes that have more vehicles will get the First Priority.

The second priority is for the Lane that has fewer vehicles than the First Lane with Priority. The regulation will be done based on the distance from DPD and waiting in front of the CPD. The timing will be regulated until all the vehicles pass the CPD sensors.

The third priority will be given to the direction NBT – SBT, while each of them has one vehicle. Therefore, we have three traffic phases:

- Phase 1 – EBT
- Phase 2 – WBT
- Phase 3 – NBT – SBT.

Based on calculations, for this Case, the entire Cycle length will be:

Phase 1 – EBT Lane. There are n_{max} = 4 vehicles, t_{L1} = 4 s. Total for the Phase is C_{L1} = 4+2*4 + 4 s = 16 s,

Phase 2 - WBT Lane. There are n_{min} = 1 vehicle, t_{L1} = 4 s. Total for the Phase is C_{L2} = 4+2*1 + 4 s = 10 s,

Phase 3 - NBT and SBT Lanes. There are n_{min} = 1 vehicle, t_{L1} = 4 s. Total for the Phase is C_{L3} = 4+2*1 + 4 s = 10 s,

In total for the Intersection, the Cycle Length for this Case is C_{TOT} = 16 + 10 + 10 = **36 s.**

Fig.16. Case 4. EBT Lane has queues of vehicles (DPD Detector Activates). Receives priority with Green Color in the traffic Light.

Case 5: All the Lanes have the same number of vehicles, 4 each, waiting in a queue. Both detectors CPD and DPD activate in each Lane. There can be no priority. There are two travel phases. This is the maximalist Case concerning the number of vehicles.

Phase 1 – EBT and WBT Lanes. There are n_{max} = 8 vehicles, t_{L1} = 4 s. Total for the Phase is C_{L1} = 4+2*4 + 4 = 16 s,

Phase 2 - NBT and SBT Lanes. There are n_{max} = 8 vehicles, t_{L1} = 4 s. Total for the Phase is C_{L2} = 4+2*4 + 4 = 16 s,

In total for the Intersection, the Cycle Length for this Case is C_{TOT} = 32 **s.**

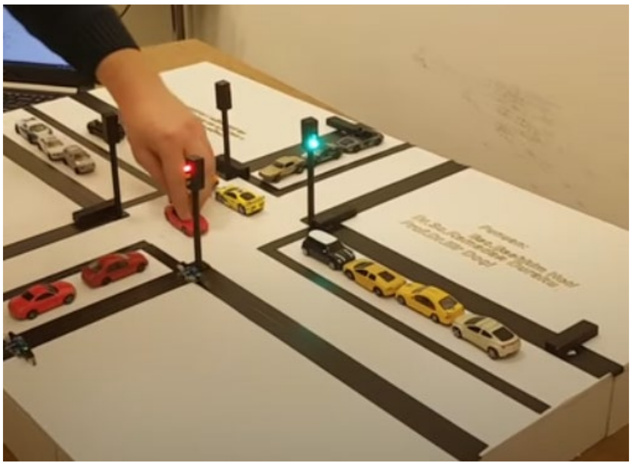

Fig.17. Case 5. All Lanes have queues of vehicles CPD and (DPD Detector Activates). There are two travel phases and no priority.

6. Performance indicators formulas and calculations

In this chapter, we will present the expressions for the calculation of the Performance Indicators that apply for this type of Intersection Model [35].

The expression for the cycle length is:

$$C = \sum_{i=1}^{n} (G_i + Y_i) \tag{1}$$

Where: G_i is actual green times for movement(s), *s*
Y_i is total length of change and clearance interval(s), *s*

When the HCM-recommended default values for l_1 and l_2 (both are= 2.0 s) are used, the lost time per phase t_{Li}, is always equal to the total of the yellow and all-red intervals Y_i [32]. Based on that, the expression for the cycle length is:

$$C = \sum_{i=1}^{n} (G_i + t_{Li}) \tag{2}$$

Where: t_{Li} is total lost time for movement(s) i, *s/phase*

It is possible to model the amount of green time required to discharge a queue of "*n*" vehicles with expression:

$$G = 4 + 2 \cdot n \tag{3}$$

Where: *n* is number of vehicles in the queue

Based on the regulation of the intersection, according to the phased principle, a load of the vehicles and the distance of the placement of the detectors, the green can be as G_{min} and G_{max}, where the expressions are also given:

$$G_{min} = 4 + 2 \times n_{min} \tag{4}$$

$$G_{max} = 4 + 2 \times n_{max} \tag{5}$$

If the start-up lost time occurs each time, a queue starts to move and the clearance lost time occurs each time the flow of vehicle stops, then for each green phase, the formula is:

$$t_L = l_1 + l_2 \tag{6}$$

where: t_L is total lost time for movement(s) i, *s/phase*
l_1 is start-up lost time, *s/phase*
l_2 is clearance lost time, *s/phase*

The total lost time per signal cycle is:

$$L = N \times t_L \tag{7}$$

Where: L is lost time per cycle, *s/cycle*

N is number of phases in the cycle, *phases/cycle*

t_L is total lost time per phase (sum of $phase_1$ + $phase_2$), *s/phase*

The number of cycles within one hour is:

$$N_C = 3600/C \tag{8}$$

The total lost time depends upon the number of cycles occurring in an hour:

$$L_H = L \times \left(\frac{3600}{C}\right) \tag{9}$$

Where: L_H is lost time per hour, *s/h*

L is lost time per cycle, *s/cycle*

C is cycle length, *s*

The number of vehicles through the intersection in one hour:

$$N_V = n \times N_C \tag{10}$$

Where: n is number of vehicles in queue, *n*

N_c is number of cycles within one hour, *cycle/h*

The remaining time within the hour is applied to effective green time for critical lane movements:

$$T_G = 3600 - L_H \tag{11}$$

Where: T_G is total effective green time in the hour, *s/h*

The following table summarizes the calculated values of the parameters.

Table 1: Calculation of intersection parameters according to the analysed cases

No.	Indicator	Case 1	Case 2	Case 3	Case 4	Case 5
1	C-cycle length, *s*	0	10	20	36	32
2	G_{min} - minimum green, *s/phase*	0	6	6	6	-
3	G_{max}- maximum green, *s/cycle*	0	12	12	24	24
4	t_L-total lost time for movement (s) i, *s/phase*	0	4	4	4	4
5	L- total lost time per signal cycle, *s*	0	4	8	12	8
6	N_c- The number of cycles within one hour, *cycle/h*	0	360	180	100	113
7	L_H- lost time per hour, *s/h*	0	1440	1440	1200	904
8	N_v- number of vehicle through the intersection in one hour, *veh/h*	0	360	720	700	1808
9	T_G-effective green time for critical lane movements, *s/h*	0	2160	2160	2400	2969

6.1. Discussions of Performance indicators results

By comparing the results presented in the Table 1, we see that each of the 5 cases has certain specifics. Comparing the cycle Length of the Case 2 (C = 10 s), and Case 3 (C = 20 s) with Case 4 (C = 36 s) and Case 5 (C = 32 s), we notice that the number of cycles per hour (N_C) decreases and, as a result, we have the reduction of time losses (L_H) per hour. On the other hand, we have the increase of the number of vehicles (Nv) that pass the intersection within the hour. The largest number of vehicles that pass the intersection are in Case 5 (Nv=1808 veh/h), which is the best result. This we will take as the main comparison parameter: compared to Case 2 it is 5 times higher, compared to Case 3 it is 2.5 times higher, and compared to Case 4 it is 2.58 times higher. Therefore, the best results are achieved for Case 5, which is the most loaded Case with vehicles. This was the aim of this work, to get the best result for the maximal loaded scenario.

According to the Literature [9], for the Intersections with Fixed Timing regulation, for the N_c = 1808 veh/h, the approximate Cycle Length is C = 42 s. In Case 5, the Cycle Length is C_5 = 32 s, which has smaller value. This is a good Result due to the application of Smart Control with Detectors.

According to these calculations, we can conclude that if the intersection is equipped with traffic signals and detectors and loaded with higher traffic flow, then the working efficiency of the traffic signals increases. In that case, the intersection parameters are improved (number of cycles, lost time per hour, number of vehicles passing the intersection within one hour, etc.). The opposite results are in the situation outside the peak hours, i.e. when we have small traffic flows in each Lane of the intersection.

7. Conclusions

In this paper, we presented the methodology for the design and regulation control of the model of Cross type Intersection. For the proper design and control of traffic models, it was important to develop accurate models that describe the real traffic processes. The method consists of 3D printing of intersection parts, calculations, mounting of parts, implementation of control hardware and programming using LabView software. Arduino Mega2560 is the selected microcontroller which enables good communication between LabView software on one side and controlled devices – sensors as detectors and Traffic Lights on other side for transmitting the signals for regulation and control.

Important part of regulation and control were programming of priorities for vehicles entering the Intersection and various cases that were planned with the algorithm. The created Model of the Intersection was tested many times to verify the functionality of parts, timings, and regulation of travel phases.

During the testing's, the model had undergone corrections in programming and hardware functionality to reach for optimal parameters of the work.

One problem in the design of these types of Intersections was the regulation of timings between travel phases and priorities. It was important to identify and regulate them to have optimal signaling regulation. Also, for the proper analysis of the functionality, performance indicators of the traffic flow were calculated.

Since this is a small prototype, there is still a lot of development potential, but the basic tasks of regulation can be performed and studied with this 3D model. This method of the design of traffic intersection model is a good example that can be implemented in other types of road intersections.

REFERENCES

1. *Lab View Software*, National Instruments Co. https://www.ni.com/en-us/shop/labview.html
2. *LabVIEW 2018 Help*, Copyright by National Instruments Corporation, 2018.
3. Arduino Mega 2560 rev3: https://store.arduino.cc/arduino-mega-2560-rev3
4. Arduino Mega 2560 Board: https://www.elprocus.com/arduino-mega-2560-board/
5. Arduino-obstacle-sensor:https://www.aliexpress.com/premium/arduino-obstacle-sensor.html?d=y
6. Mike Slinn, Paul Matthews, Peter Guest, *Traffic Engineering Design, Principles and Practice*, 2005, Elsevier Ltd.
7. Prof.dr. Ilir Doçi, *Sistemet informative të operatorëve të rrjetit (Information systems for Road network operators) (teaching Book)*, University of Prishtina, Faculty of Mechanical Engineering, Program of Road Traffic, Prishtina, 2015.
8. Jorge de Freire de Sousa and Riccardo Rossi, *Computer-based Modelling and Optimization in Transportation*, ISBN: 3319046292, Springer.
9. Salter R.J. (1996) Optimum cycle times for an intersection. In: Highway Traffic Analysis and Design. Palgrave, London. https://doi.org/10.1007/978-1-349-13423-6_36
10. Jaume Barceló, *Fundamentals of traffic simulation,* ISSN 0884-8289, Springer Science+Business Media, LLC 2010.
11. A.D'Ambrogio, G.Iazeolla, L.Pasini, A.Pieroni, *Simulation model building of traffic intersections*, Simulation Modelling Practice and Theory, Volume 17, Issue 4, April 2009, Pages 625-640.
12. Lebacque J-P., Mammar S., Haj Salem H., *An Intersection Model Based on the GSOM Model*, IFAC Proceedings Volumes, Volume 41, Issue 2, 2008, Pages 7148-7153.
13. J. Fabianova, P. Michalik, J. Janekova, M. Fabian, *Design and evaluation of a new intersection model to minimize congestions using VISSIM software*, DE GRUYTER 2020; 10:48–56.
14. Jakob Erdmann, Daniel Krajzewicz, *SUMO's Road Intersection Model*, © Springer-Verlag Berlin Heidelberg 2011.
15. Ilir Doçi, Musli Bajraktari, *Studying the infulence of heavy transportation vehicles in congested urban traffic network using traffic simulations,* international journal - MTM, Bulgaria, Page 29-32; ISSN 1313-0226, 2011.
16. Ilir Doçi, Musli Bajraktari, S*tydying the effects of right turn in congested urban intersections using traffic simulations,* International Research/Expert Conference - TMT 2011, Prague, Czech Republic, 2011.

17. Krasniqi, R., Doçi, I., Shala, A., Berisha, R., *Regulation of traffic flow in small cities with high number of vehicles: Case of Malisheva city – Kosovo*, International Review of Civil Engineering *(IRECE),* Vol 9, No. 4 (2018), ISSN: 2036-99132018.
18. Manan Temani, Rohank Agarwal, Rhythm Kohli, Mrinal Sen, *Smart Approach to Traffic Management using LabVIEW*, Fifth International Conference on Intelligent Systems, Modelling and Simulation, 2014.
19. N. Dinesh Kumar, G.Bharagava Sai, K.Shiva Kumar, *Traffic Control System using Labview*, Global Journal of Advanced Engineering Technologies, Vol2-Issue2-2013 ISSN: 2277-637047.
20. Rajkumar L Biradar, Phanindra Reddy K., Girija Vani g., *Prototype Implementation of Density Based Traffic Light Control System Using NI LabVIEW and NI myDAQ*, IJATIR, Vol.07, Issue.16, November-2015, ISSN 2348–2370.
21. Nicolae Patrascoiu, Cosmin Rus, *Traffic signal controller implemented through a virtual instrument*, 2019 15th International Conference on Engineering of Modern Electric Systems.
22. Srinivas B, S. Tata Reddy, S. Guru Swaroop, P. Indumathi, *Density based Traffic Control System using Ni LabVIEW*, JASC: Journal of Applied Science and Computations, Volume 6, Issue 3, March/2019, ISSN NO: 1076-5131.
23. Ankit Dhiman, Atul Kumar, Ajay Kumar, *Modeling of Traffic Light Control Using LABview*, International Journal of Scientific Research in Computer Science, Engineering and Information Technology © 2017, IJSRCSEIT, Volume 2, Issue 6, ISSN : 2456-3307.
24. Prof.dr.Ilir Doçi, Mr.sc. Ramadan Duraku, *Detyra të zgjidhura me softverin LabView* (*Solved Examples with LabVIew Software),* Teaching Book, Prishtina, 2018.
25. M. Keanu, J. E. Exconde, J. Anne A. Co , J. Z. Manapat , E R. Magdaluyo, Jr, *Materials Selection of 3D Printing Filament and Utilization of Recycled Polyethylene Terephthalate (PET) in a Redesigned Breadboard*, Procedia, CIRP 84 (2019) 28–32.
26. N. Shahrubudina, T.C. Leea, R. Ramlana*, An Overview on 3D Printing Technology: Technological, Materials, and Applications*, SMPM 2019, Procedia Manufacturing 35 (2019) 1286–1296.
27. Richard Jenning, Fabiola de la Cueva, *LabVIEW Graphical Programming, Fifth Edition*, McGraw-Hill Education, 2019, ISBN-13: 978-1260135268.
28. LINX, LabView MakerHub: https://www.labviewmakerhub.com/doku.php?id=libraries:linx:start
29. *LabVIEW 2020 Community- Ni-DAQMX Help*, Copyright by National Instruments Corporation.
30. *Interfacing LabVIEW with Arduino*, Interfacing (USB – Rs232 – I2C -ISP) Projects, https://duino4projects.com/interfacing-labview-with-arduino/
31. R. Singh, A. Gehlot, B. Singh, S. Choudhury, *Arduino-Based Embedded Systems, Interfacing, Simulation, and LabVIEW GUI*, CRC Press, © 2018 by Taylor & Francis Group, LLC.
32. *Highway Capacity Manual 2010*, Transportation Research Board, National Academy of Sciences, 2010.
33. UP-FME labs: https://fim.uni-pr.edu/Laboratoret/Mekatronike.aspx
34. Additional NI Product Boards, LINX Community Platform, https://www.labviewmakerhub.com/doku.php?id=libraries:linx:start
35. Roger P. Roess, Elena S. Prassas, William Mc. Shane, *Traffic Engineering,* 3rd Edition, 2004.
36. 3D Printing Filament for Design and Manufacturing, https://www.raise3d.com/filaments/
37. NI LabVIEW LINX Toolkit, https://sine.ni.com/nips/cds/view/p/lang/en/nid/218410