Traffic Control Simulator - Final Report

Group 8 - João Diniz, José Sousa, Luís Negrão, Sara de Sá

I. Objectives of the Project

The objective of this project is to develop a system capable of controlling traffic lights in order to reduce queues of cars and pedestrians, the waiting time, and also the waiting time on the displacement of people.

The design implemented must be able to locate rows of cars that are simulated with buttons and assign a higher priority to longer queues (with 10 cars or more) without causing hunger in the shortest lines.

Another objective of this project are the traffic lights for the pedestrians, which can interact with the remaining traffic lights and can use a button to request a green light to cross.

Finally, the system had to be able to represent a simulation of the system on the smartphone of a user and he could be able to change the light signals as long as it was not breaking the rules.

II. Project Requirements

For this project it's defined that we use three different types of systems,Arduino, Raspberry Pi and Android to model a traffic control system.

We have certain traffic requirements that we need to fulfill, which are the following:

- at least one crossing, with pedestrian lights in at least one street.
- cars should not wait more than 15 sec on a red light.
- system should detect congestion (more than 10 cars stopped at a light) in less than 2 sec.
- queues with 10 or more cars should not have the red light on for more than 10 sec.

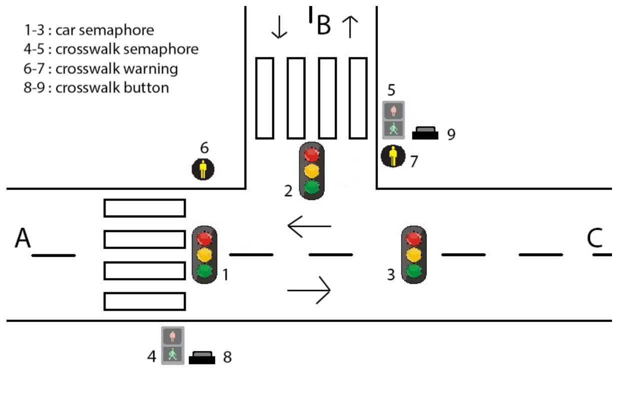

Fig. 1. Traffic system

For the previous stated requirements, and the chosen traffic system, we will need at least two Arduinos, a Raspberry Pi, Android phone, a set of light to represent the traffic lights and buttons to emulate the car build up on the roads and for the crosswalk buttons.

III. Project Development

The project was develop to work with three different types of systems, Arduino, Raspberry pi and Android. Each system has different requirements and functionalities.

Previously we estimated the overall cost of the project, however, we ended using more hardware, than stated. In the Table II we show full list of hardware used, and in table I we do the cost of each component.

TABLE I
COMPONENTS ESTIMATED COST

Component name	Component cost in Euros
Semaphore	2.5
Arduino Mega 2560 rev3	31.40
Raspberry Pi 4	74.04
Red Led	0.40
Yellow Led	0.20
Green Led	0.40
Button	0.15
Wires	0.02
Resistance 330 ohm	0.06
Resistance 4.2k ohm	0.50
Mini Breadboard 170 ponts	1.88
USB A-B Cable	1.90
Samsung S4 mini	248

We can estimate the full cost of the project to be 390 euros.

Further in this section we will explain the objectives and implementations of each system.

A. Arduino

The Arduino is responsible for handling all the low level components, i.e. the light and the buttons. The Arduino job is to receive commands from the raspberry pi and inform about

TABLE II
COMPONENTS USE

Component name	Number of Units
Semaphore	3
Arduino Mega 2560 rev3	1
Raspberry PI 4	1
Red Led	2
Yellow Led	2
Green Led	3
Button	5
Wires	30
Resistance 330 ohm	18
Resistance 4.2k ohm	5
Mini Breadboard 170 points	3
USB A-B Cable	1
Samsung S4 mini	1

the state of each components, i.e. indicates when a certain light suffered a change or when a button was pressed. Is the responsibility of the raspberry pi to manage the logic of the system. The Arduino never talks directly with the Android component.

The Image 2 show an example of the circuit constructed to manage one street. The actual circuit uses an Arduino Mega to manage three different roads, but basic idea maintains.

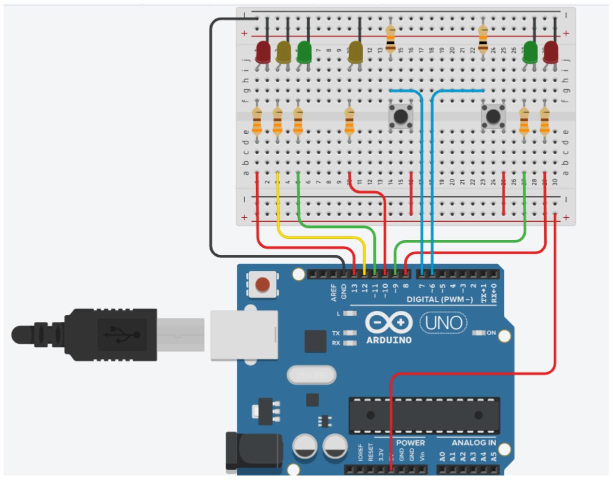

Fig. 2. Example of a layout for one street

1) Communications between Arduino and Raspberry pi: The Arduino currently has the capacity of receiving and sending messages to Raspberry pi, using a serial port. To achieve this task, a simple protocol was made. The Table III-A1 shows the size and fields of the messages exchanged between the Arduino and Raspberry pi.

1 Byte	1 Byte	2 Byte	1 Byte	1 Byte
TP	TO	ID	State/Req	End Sym

Where:

- TP - Is the type of message. S for the state of the components and C to change the state of a given component
- TO - Type of object, e.g. T for the traffic lights, P for the pedestrian lights, p for the pedestrian buttons, t for the traffic buttons and W for the warning lights.
- ID - Identifies a given component
- State/Req - To inform of change in a given component, or to inform the current state of a component.
- EndSym - The terminator symbol, new line

As stated before, the raspberry pi is now in charge of running all the logic of the system

B. Raspberry Pi

The Raspberry Pi is responsible for handling the logic of the system,i.e maintaining the state machine algorithm that controls the traffic lights(the buttons). Contains an internal model of the entire crossing on which it will act taking into account the information provided by the Arduino system. The all implementation of the Raspberry Pi is done with Java language.

C. Android

The Android system collects the information about the traffic lights and try to visualize their conditions. Also it can enable user input on the traffic lights to request change of the traffic lights on the crosswalk and the number of cars waiting on the red light.

1) Communication between Android and Raspberry pi: The communication between the Android and the RaspberryPi is done over wifi, and they are able to send and receive responses from each other and its done through 2 sockets.

One of them is responsible for receiving the state changes that occur in the Arduino and the other is responsible for allowing Android to make state change requests.

In the socket that communicates with the Arduino, what happens is that the system receives an integer 0, 1, 2 or 3 according to the number corresponding to the current state.

In the socket that communicates with Android, the mobile phone sends the integer 1 if it wants the system to advance to the next state and sends 0 if it wants the system to return to the previous state.

IV. Problems found and their solutions

In this section we will discuss some of the problems faced in making the proposed project.

A. Arduino

The Arduino was very straight forward, the only two problems faced was in getting the hardware, since the faculty had some trouble finding the whereabouts of the buttons, and the input of the buttons, that were leaking current, somehow, however, with the help of the professor we were able to avoid the issue, by changing the circuit configuration.

B. Raspberry Pi

When we first started to implement the Raspberry Pi, we tried to communicate with the Arduino and we had some problems,but they were fixed by introduction individual settings in the code.

After that the raspberry pi implementation itself was very straightforward, we managed to complete all the proposed requirements for this system.

The only final problem that we had was in relation to the visualization in Android windows, since when we switch from one window to another, the previous one stops working.

We believe that it is not a communication problem between the 2 systems, as it continues to happen and to show the results in the terminal, only the visualization fails.

C. Android

The android implementation was more difficult to execute than the other 2, because we chose to develop it with the Kotlin language. Since this language has a recent updated version of its documentation, there wasn't much information on it to rely on. In addition, the communication between Android and the Raspberry Pi, we also encountered some problems in the case of changing the lights of the traffic lights (because when we changed windows between the one that stores the state in real time and the one that controls the state, the previous one left to work). Despite these problems that appeared, we managed to solve them, and the proposed requirements for the system were all concluded.

V. Achieved Objectives

In this project, we were able to emulate a traffic control system with the requirements that were proposed. Our system has at least one crossing (in our case we have 2 crossings with pedestrian lights in both of them). We guarantee that the cars don't wait more than 15 sec on a red light, the queues with 10 or more cars don't have the red light for more than 10sec.

VI. Not Achieved Objectives

With the time that we had for the development of the project, we were able of achieving all the objectives that we had defined in the initial report. We just have a little problem with the visualization on the Android, as we mentioned in the section of the Raspberry Pi in this report.

All of the code implemented for this project is available at our git repository: https://github.com/EnioSousa/SE-Project