Instituto Superior Técnico

Applications and Computation for the Internet of Things

2022-2023

3rd Lab work – Project: Control of Traffic Lights at a Roundabout[1]

Goal

The goal of this project is the implementation of a modular and automatic traffic lights system for a roundabout. The overall system has some degree of fault tolerance and fail safe behavior.

In the laboratory traffic lights will be replaced by LEDs, otherwise the system to be implemented exhibits realistic behavior. All traffic lights are connected to a central controller via an I2C bus.

Description

The purpose of the project is the design and development of a traffic lights system to control traffic at a roundabout with four accesses.

Each access supports the following movements of vehicles:

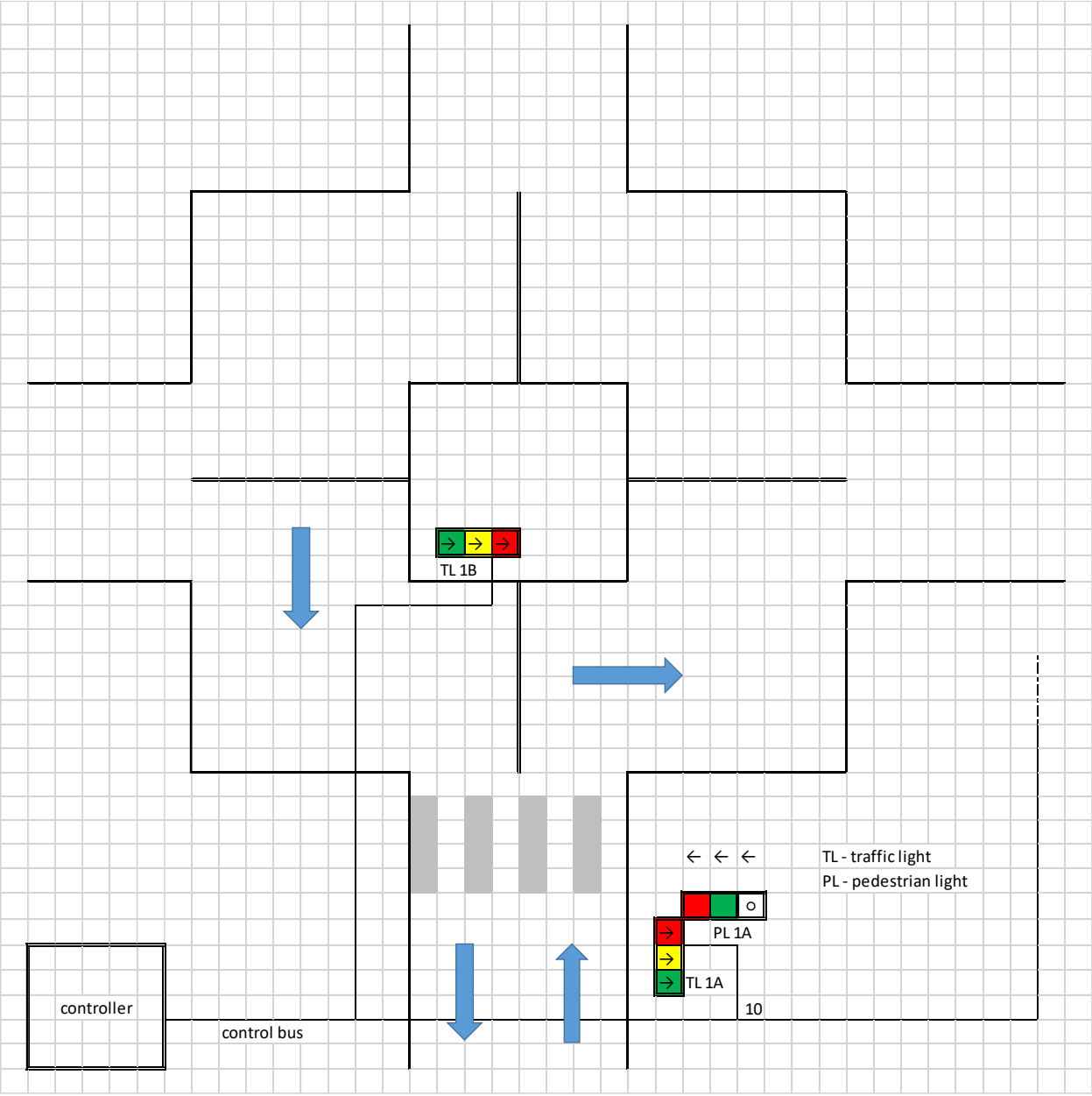
- For vehicles entering the roundabout: Turn right
- For vehicles in the roundabout:
    - Continue in the roundabout
    - Get off the roundabout turning right

Each access – k – has two traffic lights (TL kA, TL kB).
TL kA controls accesses of vehicles to enter the roundabout, and also has lights and a press button for pedestrians to signal their intent to cross the street (PL kA). (To simplify the assembly of the circuit only one set of pedestrian lights and button will be implemented on one side of the street.)

[1] V2.2, 21 December 2022. V2.1, 12 December 2022.

TL kB controls the flow of vehicles in the roundabout, preventing the passage of vehicles once traffic in the nearby street is allowed in the roundabout. Only vehicles turning right, to get off the roundabout, are allowed to proceed.

All traffic lights are connected through an I2C bus to a central controller which orchestrates all traffic lights.
The functionalities of the controller and of the traffic lights must be implemented according to a specified interface protocol so that any traffic light respecting the interface can work with any implementation of the controller. (The traffic light implemented by a group of students must be interoperable with a controller implemented by another group.)



Traffic Light

The traffic light is implemented with an Arduino UNO with multiple I/O devices. Every traffic light must have up to 5 LEDs: 3 of them represent the traffic control color signals (TL kA) – green, yellow, and red – and 2 represent the pedestrians control signals (PL kA) – green, and red. Every traffic light may also have a pedestrian button to reduce the waiting time for pedestrians to cross the street. Each traffic control light (green, yellow, and red) must have fault detection capability to detect that it is always turned OFF and does not react to its control.

To simplify the implementation of the prototype each Arduino board may control both traffic lights corresponding to an entry in the roundabout (TL kA, and TL kB). However the control tasks associated with each traffic light must be implemented with self-contained software modules, exchanging information through a strict message passing discipline, without any variables in shared memory. The coordination between TL kA and TL kB must be self-sufficient, implemented locally and not by the roundabout controller.

The identification of the roundabout entry corresponding to the traffic light is configured by jumpers (or fixed wires) connected to input ports of the Arduino controller – 1, 2, 3, 4 – in ascending order anti-clockwise.

Controller (Orchestrator):

The controller must be designed as an Arduino UNO with I/O devices attached.

- 2 LEDs: a red LED to show the controller status (ON or OFF), and a blue LED to indicate activity on the communications bus;
- an ON/OFF button (turn on/turn off);
- a potentiometer to select the period of traffic control (enter street #1 → enter street #2 → enter street #3 → enter street #4 → enter street #1).

To reduce the hardware requirements the controller function may be implemented on an Arduino UNO board associated with an entry of the roundabout.[2] However it must be implemented by a self-contained software module, exchanging information with the traffic lights function through a strict message passing discipline, without any variables in shared memory.

[2] I. e. the controller/orchestrator function shares the same hardware controller with a traffic light function. The API of a controller, to send/receive messages to any other controller, follows a message-passing discipline. It is the responsibility of the message-passing routine to check if messages need to be sent through the I2C or can be exchanged within the context of a processor.
The only exception from this strict discipline is the communication between the two traffic lights in an entry. To guarantee local failsafe, independently of the global infrastructure, the controller of entry I will manage both traffic lights TL_kA, TL_kB.

The identification of the controller node is ~~configured by jumpers (or fixed wires) to 0 (zero)~~.[3]

Requirements

1. Initial state of the system must – controller turned OFF (red LED OFF); all traffic lights blinking yellow with a 1 second period (ON + OFF cycle time).
2. It must be possible to turn the controller ON and OFF, pressing the button:
    a. When turned ON, the controller must start a cyclic sequence of control of the roundabout. Starting with entry 1:
        i. Block entries 2, 3 and 4 - Command entries 2, 3 and 4 to go RED, wait for the acknowledgements, command entry 1 to go GREEN.
        ii. Wait a control period [2, 15] seconds (controlled by potentiometer).
        iii. Block entry 1 – command entry 1 to go RED, wait for the acknowledgement, command entry 2 to go GREEN, and so on (then 3, 4, 1, …).
    b. When turned OFF, the controller must signal all traffic lights to start blinking yellow, going back to the initial state.
3. While receiving or sending data the controller's blue LED must blink.
4. The traffic light color transitions must be:
    a. Red → Yellow (immediately before: pedestrian Green → Red).
    b. Yellow → Green.
    c. Green → Yellow.
    d. Yellow → Red (immediately after: pedestrian Red → Green).
    e. Each passage through Yellow will take 0,5 seconds (constant, independent of the control period).
5. It must be possible to shorten the cycle time by half by pressing the pedestrian button - when the button is pressed the remaining of the cycle time is halved. The reduction affects a single cycle after which the system reverts to its normal operation cycle.
6. While one traffic light is performing a red-yellow-green-yellow-red cycle, the other must have its red light always on (and, in part, pedestrian green). In other words, two traffic lights must not be simultaneously green.
7. All communications between the traffic lights must be performed via controller (see section Modular Programming).
8. It must be possible for the controller and any traffic lights, to detect faults of the communications link, or in other traffic lights:
    a. After 2 control periods – [2×2, 2×15] seconds – with persistent faults:[4]

[3] We consider that the orchestrator can be installed on a processor together with a traffic lights controller. To simplify the implementation assume that the orchestrator will be installed together with entry controller #1, therefore:

- the I2C interface of this processor must be able to exchange messages from/to nodes 0 and 1;
- it is not necessary to configure the location of the orchestrator "with jumpers to 0 (zero)", since the address of this node will be '1'.

[4] The protocol changes suggested change and simplify the detection of faults: "Removed all the messages sent from the traffic lights. Now traffic lights will just answer to controller requests." Now the

i. The traffic light that detected the missing or faulty communication must start blinking yellow (both lights which control traffic at the entry – TL kA and TL kB).
ii. The controller must communicate to other traffic lights to start blinking yellow, too. Then it must turn itself off, going back to the initial state of the system.
iii. Faults can be detected by the controller (e. g. lack of response to commands from a traffic light), or by a traffic light (e. g. lack of command from the controller to change the accesses to the roundabout).

9. All communications in the system must be performed using I2C protocol.

Modular Programming

Communications between the modules of the system (traffic lights and controller) must respect a specific API in order to build a modular and interoperable system. This approach:

- will enable two teams of programmers, working in parallel, to build both controller and traffic light separately, and
- will allow controllers and traffic lights from different teams to work together properly.

Controller - Traffic light communication API:

Controller – The controller must implement the following messages to be sent via I2C:

- RED (x) – where X is the identifier of the entry to be closed.
- GREEN (x) – where X is the identifier of the entry to be opened.
- OFF (x) – signals a traffic light to start blinking yellow.
- PING (x) – where X is the identifier of the entry to be checked. It requests the traffic light of the entry to send back its status. (In this case the STATUS replaces the ACK message.)
- ACK (x) – It is the acknowledge message returned in response to x. The request/command was correctly received.

Traffic lights – traffic lights must implement the following functions to be sent via I2C:

- TIME (x) – signals the controller (x = 0) to shorten the cycle period due to a pedestrian request.
- PING (x) – where X is the identifier of the entry to be checked. It requests the traffic light of the entry to send back its status. (In this case the STATUS replaces the ACK message.)
- STATUS (x, state) – Sends to x the status of the traffic lights at the entry.
- ACK (x) – It is the acknowledge message returned in response to x. The request/command was correctly received.
-

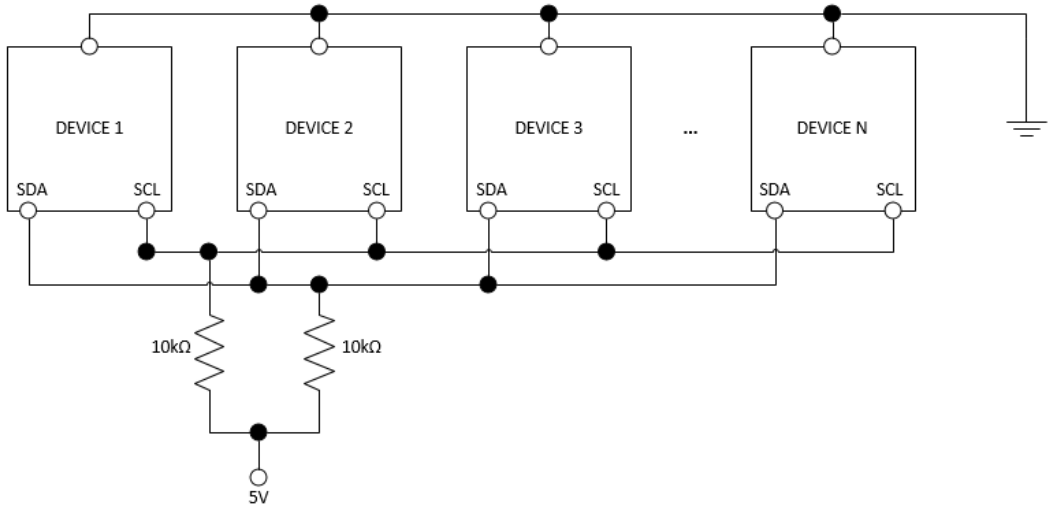
Check the data interchange specification for a detailed description of the commands.

---

controller (orchestrator) detects a faulty traffic light controller, instructs it to start blinking (if possible), and also instructs the other traffic light controllers to do so.

I2C with Multi Master-Slave diagram

To build a multi Master-Slave I2C circuit for the Arduino UNO students must implement the following circuit.



Simplified Input/Output

When merging several modules on the same Arduino controller the number of interface ports required to implement the specifications may exceed the number of input/output pins available. Therefore the following simplifications are recommended.

Requirement	Simplification
Controller: two LEDs to show status and communications activity.	Use only one LED to show both status (ON/OFF) and communications activity, eventually pulsing with different patterns.
Identification of the Arduino controllers: • 0 – controller; • 1, 2, 3, 4 – pairs of traffic lights at each entry of the roundabout.	Code the identifier of each Arduino controller with just two bits: • 1, 2, 3, 4 – pairs of traffic lights at each entry of the roundabout; • The main controller of the roundabout is loaded at Arduino nº 1.
Traffic Light: check the operation condition of the 3 lights (red, yellow, and green).	Check only the operation condition of the red LED since it is the most critical to enforce safety.
Traffic Light with pedestrian signal.	Implement only the green pedestrian signal: green ON means “safe to cross the street”, green OFF “do not cross, or take care to cross”.

Recommendations

In order to fulfill your work with security and not damaging the hardware involved, remember to carry out the recommendations below. As you are working fill the boxes to be certain that you fulfill all security measures.

Always work with the circuit disconnect from the source.	
Call the professor or responsible for the laboratory, before you connect the circuit to the source.	
Make sure the circuit is well connected (resistors, capacitors, etc.) to prevent a short circuit, or damage the hardware.	

Plan and Deliverables

Week		Activity
1	Dec 19	Project presentation. Requirements analysis and specification of the data interchange formats of I2C messages.
2	Jan 2	Project design and development.
3.a	Jan 9	*First half of the shift.* Delivery of the project report (printed report, and e-mail), including a. design of the circuits of the controller and traffic lights, b. overall architecture of the software in the controller and traffic lights, c. safety and fault-tolerance measures adopted, d. programs (properly commented) implementing the controller/orchestrator and traffic lights. (The report includes interoperable functionalities to be demonstrated in session 3.b.) Demonstration of traffic lights of an entry (TL kA + TL kB + PL kA) by each group.

Week		Activity
3.b		*Second half of the shift.* Demonstration of the integrated system controlling the roundabout by sets of student groups to be defined. (Each set will have 2 groups of students, each group handling two entries of the roundabout. Example: group A controls entry 1 with the main controller, and entry 3, group B controls entries 2 and 4.) Each set of two groups delivers (by e-mail until the end of the day) a brief evaluation report (one A4 page) of the integration test, to complete, correct, or confirm the reports delivered at the beginning of session 3.a.
4	Jan 16	Oral evaluation of the groups. Covers all laboratory works and the project.