

# How To MODIFY this project

---

## 1. Modify the software

### 1.1. Modify the Arduino code

The Arduino code is located in the folder `arduino_script`. It is composed of 2 files:

#### 1.1.1. `arduino_arena_receiver.c`

This file is the main file. It is the one that is uploaded to the Arduino. It is composed of a main loop that is executed indefinitely and choosing between 4 states:

- 'Opening the 1st barrier'
- 'Opening the 2nd barrier'
- 'Closing the 1st barrier'
- 'Closing the 2nd barrier'

The traffic lights are linked to the barriers so that they are green when the barrier is open and red when the barrier is closed. Those functions are implemented in this file. To modify the behavior of the traffic lights, you have to modify the functions `open_barrier()` and `close_barrier()`.

#### 1.1.2. `communication_arduino.py`

This file is the one that is executed on the Raspberry Pi. It allows the communication between the Arduino and the Raspberry Pi. It is composed of 2 functions:

- `callback()`: This function is called when a message is received on the topic `arene/cmd`, it makes the Arduino execute the command received.
- `send_to_arduino()`: This function sends a number to the Arduino on the serial port. This number is the command that the Arduino will execute. The main function of this file initializes the node and the subscriber to the topic `arene/cmd`.

### 1.2. Modify the launch files

The launch files are located in the folder `launch`. There are 4 launch files:

#### 1.2.1. `arene_real.launch`

This launch file is used to launch the project on the real arena. It launches the node `communication_arduino.py` and an example of a node that sends commands to the Arduino. To modify, remove the node that sends commands to the Arduino and replace it with your own node.

#### 1.2.2. `arene_simulation.launch`

This launch file is used to launch the project on the simulation arena. It launches Gazebo and initializes the world with the arena and the turtlebot. It contains the `world_control` node that sends commands to the Arduino automatically. There is also an example of teleop and world control nodes. To modify, remove the

`world_control` node and replace it with your own automatic control node or remove the `example_control` node and replace it with your own control node.

### 1.3. Modify the scripts

The scripts are located in the folder `scripts`. There are 4 scripts:

#### 1.3.1. `world_control.py`

This script contains 2 classes that are used to control the modules, the `Barrier` class and the `TrafficLight` class. A `Barrier` contains a `TrafficLight` this way they are linked together. For each class there is a method `change_state()` that changes the state of the module. The different state are open and closed for the barrier and green and red for the traffic light. The method `change_state()` can be modified to add a blinking yellow light for the `TrafficLight` for example as for the barrier you can control the speed of the barrier. You can try to modify the part that changes the position of the barrier but you should keep the other parts that are making it work altogether. If you add new modules you must update the object list in the main function of the script. To modify the message on the topic `arene/cmd` you should modify the function `callback()`.

#### 1.3.2. `example_control.py`

This file is an example of a control node that you can use to control the arena.

#### 1.3.3. `teleop_gazebo.py`

This file is an example of a teleop node that you can use to control the turtlebot in the simulation arena.

#### 1.3.4. `checkup_turtlebots.py`

This file was used to check if the turtlebots were operational. It can be malfunctionning if used outside of a virtual machine, especially the matplotlib library.

## 1.4. Modify URDF Files for modules

The URDF files are located in the folder `urdf`. There are 2 URDF files:

#### 1.4.1. `barrier.urdf`

This file is the URDF file for the barrier. It is a very simple URDF file that contains a base for the barrier and the moving part of the barrier. You can add several more parts but keep in mind to adapt your control node that apply the force to the moving part of the barrier. Do not forget to add dynamics settings to your parts.

The link `world` allows the barrier to be static in the world, you have to remove it if you want to move the barrier around. Removing it also means that the barrier will fall after moving or colliding with something.

#### 1.4.2. `traffic_light.urdf`

This file is the URDF file for the traffic light. It contains a base and 3 moving parts for the 3 lights. You can modify this design, for example by making it vertical with a rotation in the joint description between the world and the base.

The link `world` makes the base fixed to a position, if you move or rotate the base you have to adapt the position of the lights and of the base in the joint.

## 1.5. Modify the world

The world files are located in the folder `world`. There are 5 files:

### 1.5.1. `arene_2.world`

This world is the one that contains the barriers, the lights and a tunnel, this is the file that was used during the project. You can modify it to add more barriers or lights or to change the position of the tunnel. You can also add more objects to the world. To do so, it is advised to open the world in gazebo, add the objects in the world and then save the world. You can ultimately, manually modify the world file to pinch the objects.

This file is quite large so here are some milestones:

- Default world state `1. 245`
- Tunnel model `1. 139`
- Barrier 1 model `1. 307`
- Barrier 2 model `1. 441`
- Traffic light 1 model `1. 575`
- Traffic light 2 model `1. 777`

### 1.5.2. `empty.world`

This world is an empty world that can be used to test the arena without any objects in it. It can also be used as a base to create a new world.

## 2. Modify the hardware

### 2.1. Modify the barrier

The barrier is made of a servo motor and a light wooden board. The servo motor is attached to the board with a screw and a nut. This module is plugged by 3 wires to the Arduino. The servo motor is plugged to the 5V and GND pins of the board directly plugged to the power supply and the signal pin is plugged to the pin A# of the Arduino.

Future modification can include a fixing system for the barrier to the ground and a more robust design for the wooden board.

### 2.2. Modify the traffic light

The traffic light is made of 3 LED and a 3D printed base. The LEDs are wired to a board with a CD4007 which consist of 3 NAND gates. The board is plugged to the Arduino with 5 wires. The 5V and GND pins are plugged to the power supply and 3 signal pins are plugged to the pins of the Arduino.

Future modification can include a smaller board, colored LEDs or colored lenses, this could allow to a smaller 3D printed base. If the field of view of the cameras used on the turtlebot is wide enough it is possible to make an aerial traffic light.