

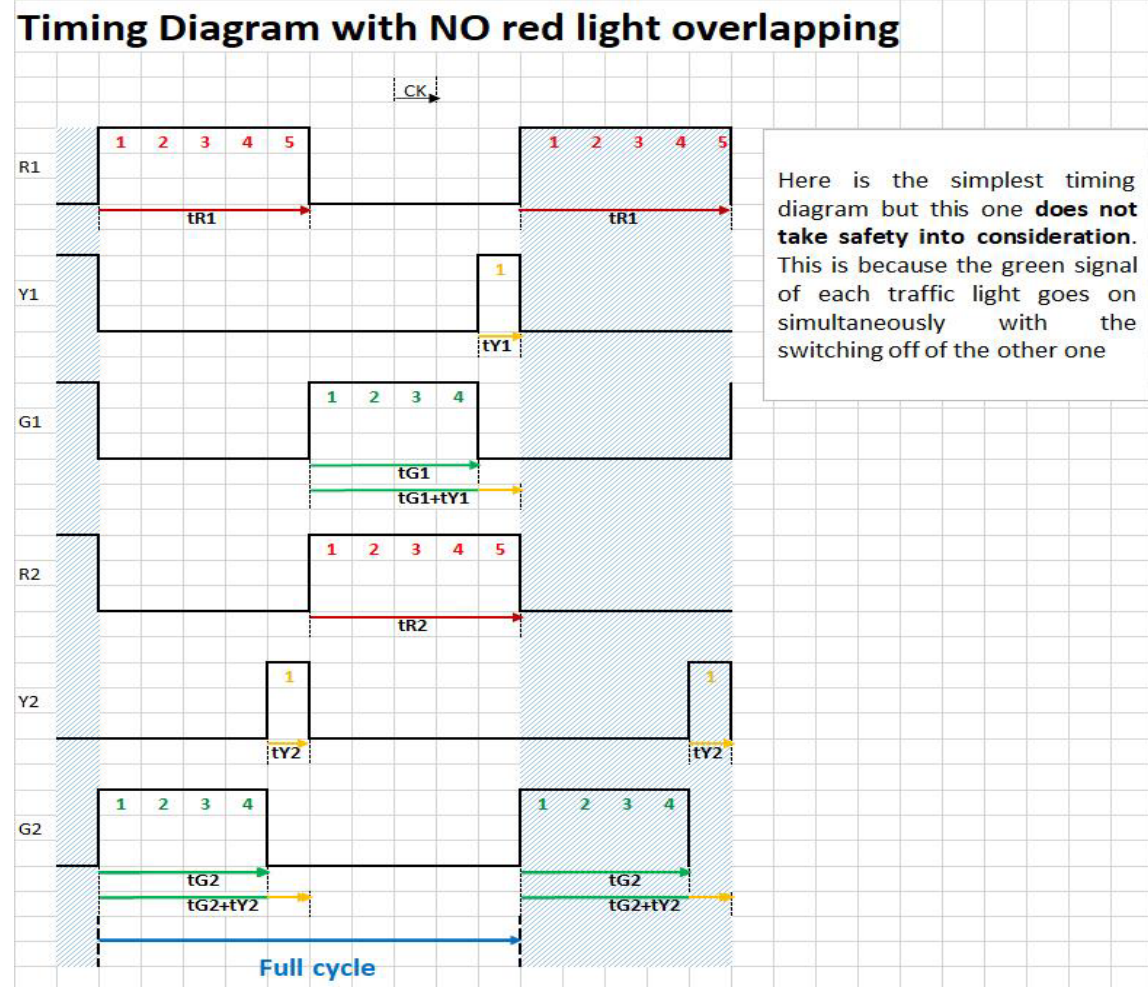
international@isisdavinci.eu



Grant number: 2020-1-PL01-KA201-081698

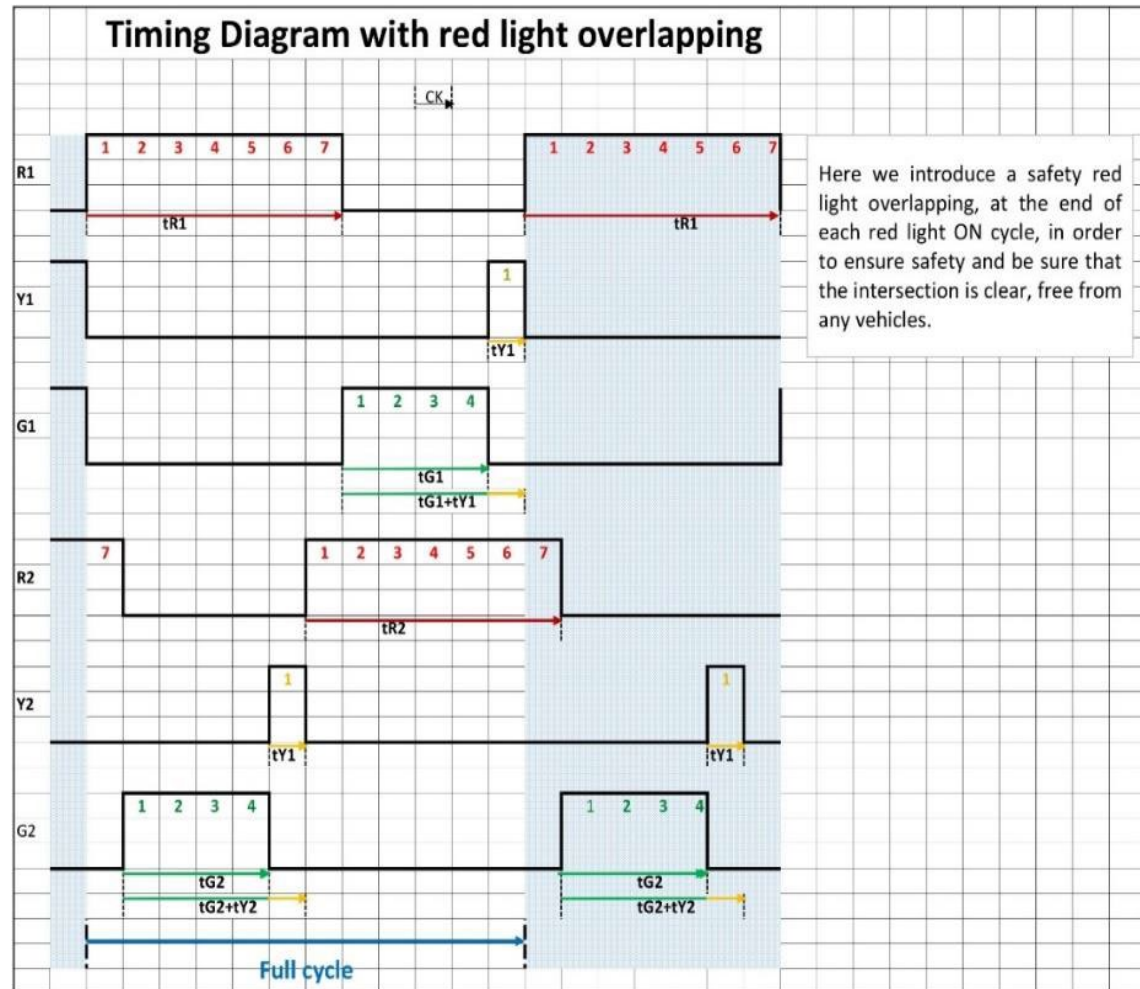
ABOUT THE PROJECT: REALIZATION OF THE TIMING DIAGRAMS

Simplest Timing Diagram with NO red light overlap



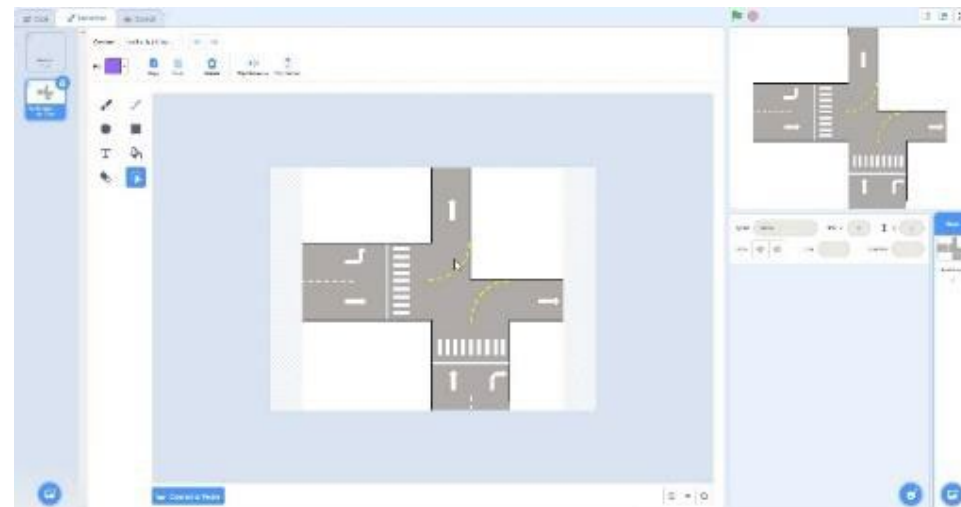
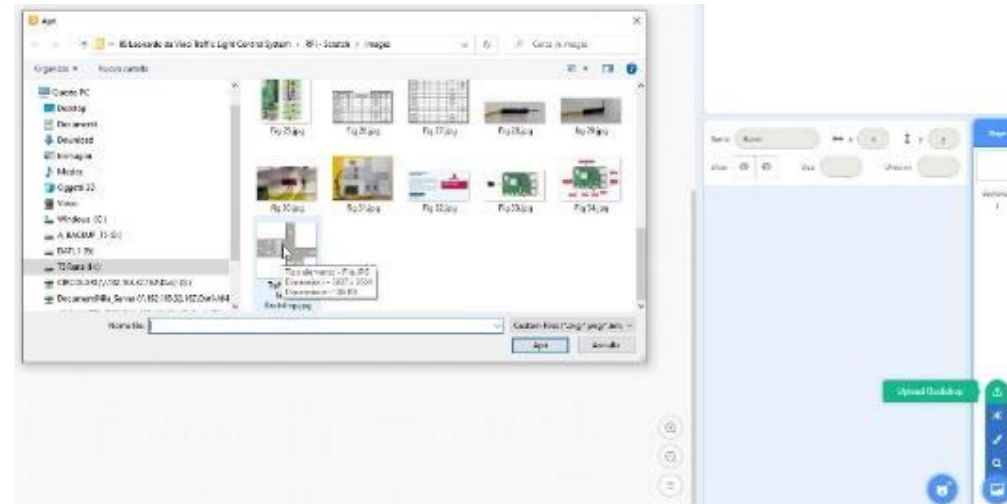
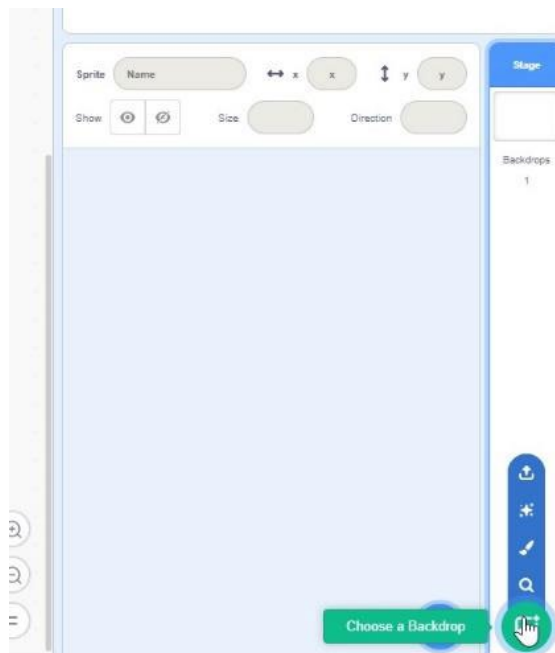
ABOUT THE PROJECT: REALIZATION OF THE TIMING DIAGRAMS

Timing diagram with red lights overlap



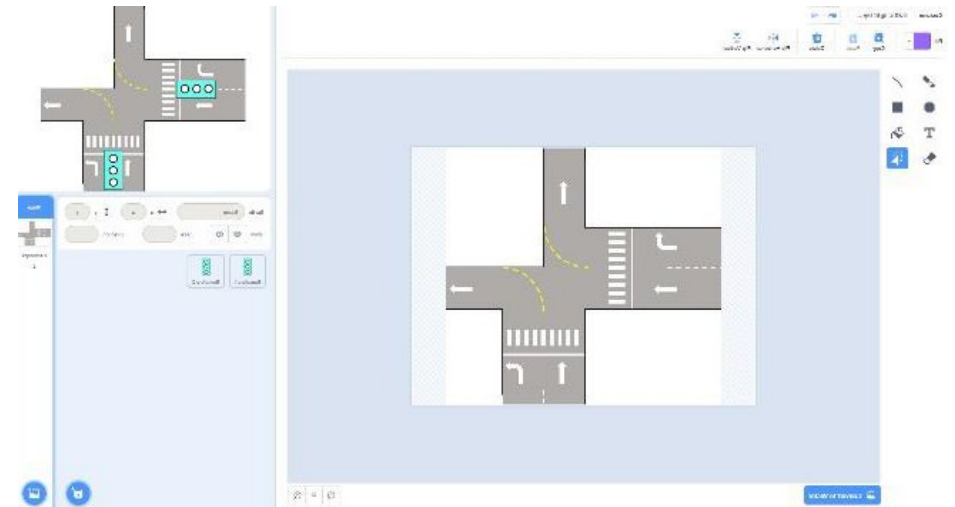
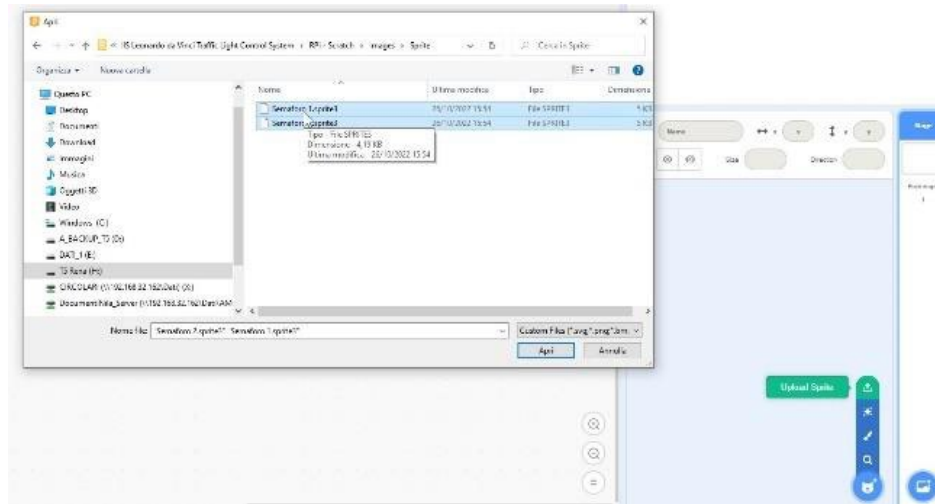
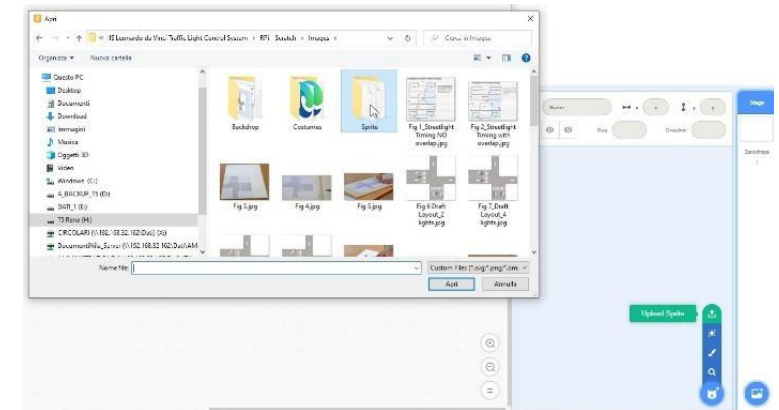
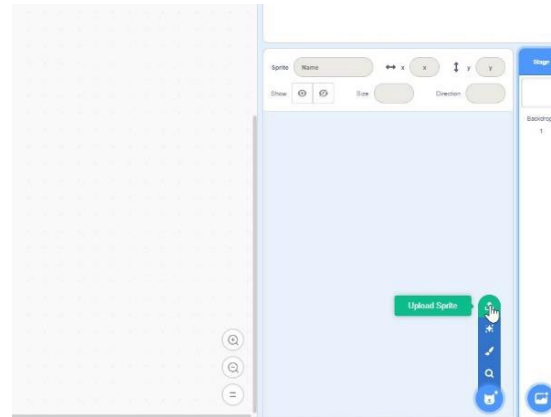
LET'S CREATE THE BASIS OF OUR FIRST "TRAFFIC LIGHTS" PROGRAM WITH SCRATCH

Create your "Backdrop" that will represent our background on which we will place the two traffic lights



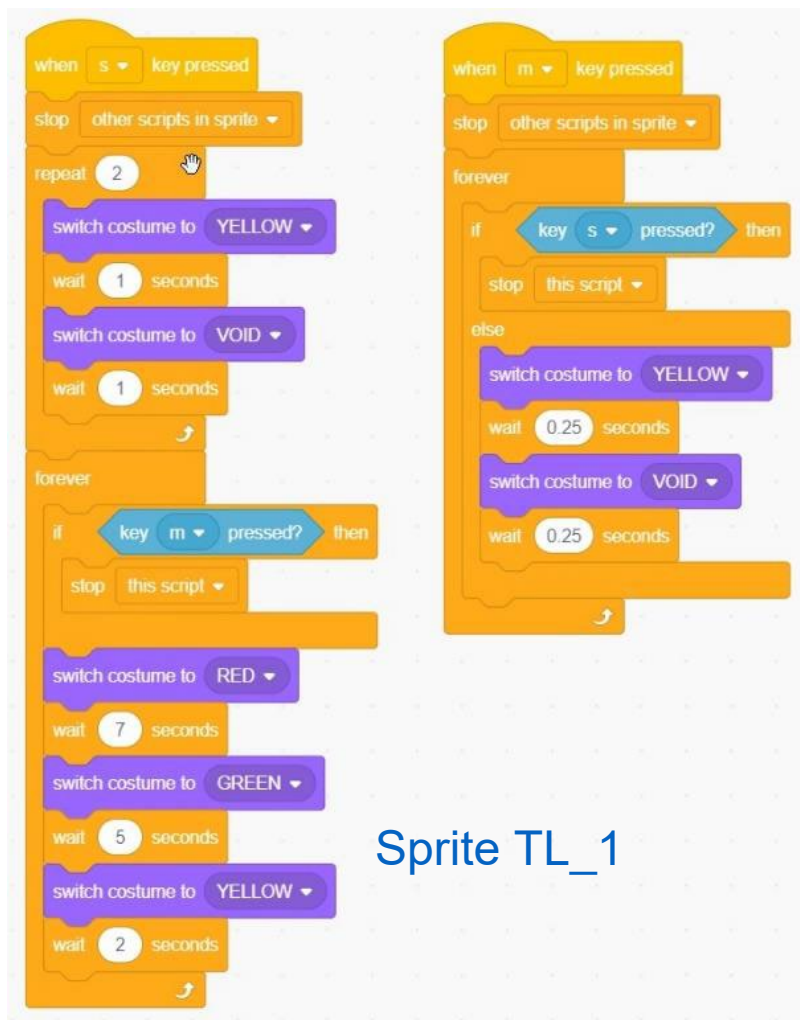
LET'S CREATE THE BASIS OF OUR FIRST "TRAFFIC LIGHTS" PROGRAM WITH SCRATCH

Create our “Sprites” that represent our Traffic Lights shapes and “Costumes”. Those are the three possible situations for each Sprite: Red, Yellow and Green

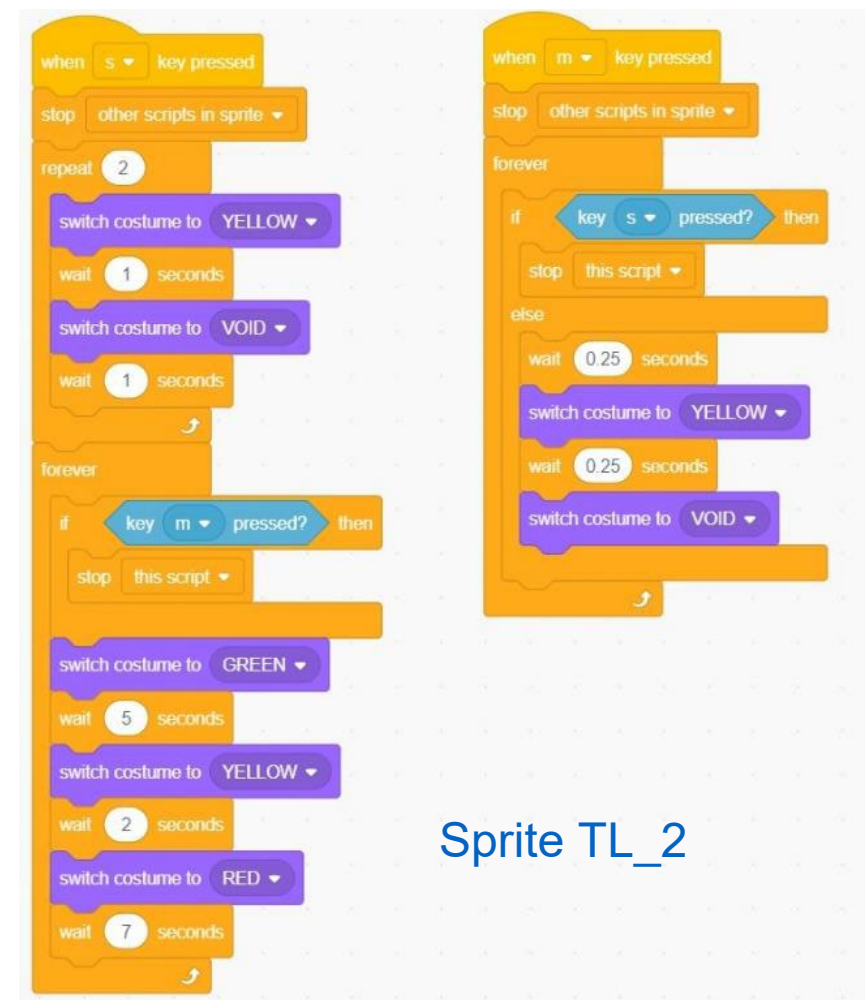


BUILD OUR TRAFFIC LIGHTS USING SCRATCH IN “DESKTOP MODE” ON THE PC

The first simple program that controls the two traffic lights **with NO red light overlap** using two keys on the PC keyboard ("s" start and "m" maintenance) to activate the two scripts:



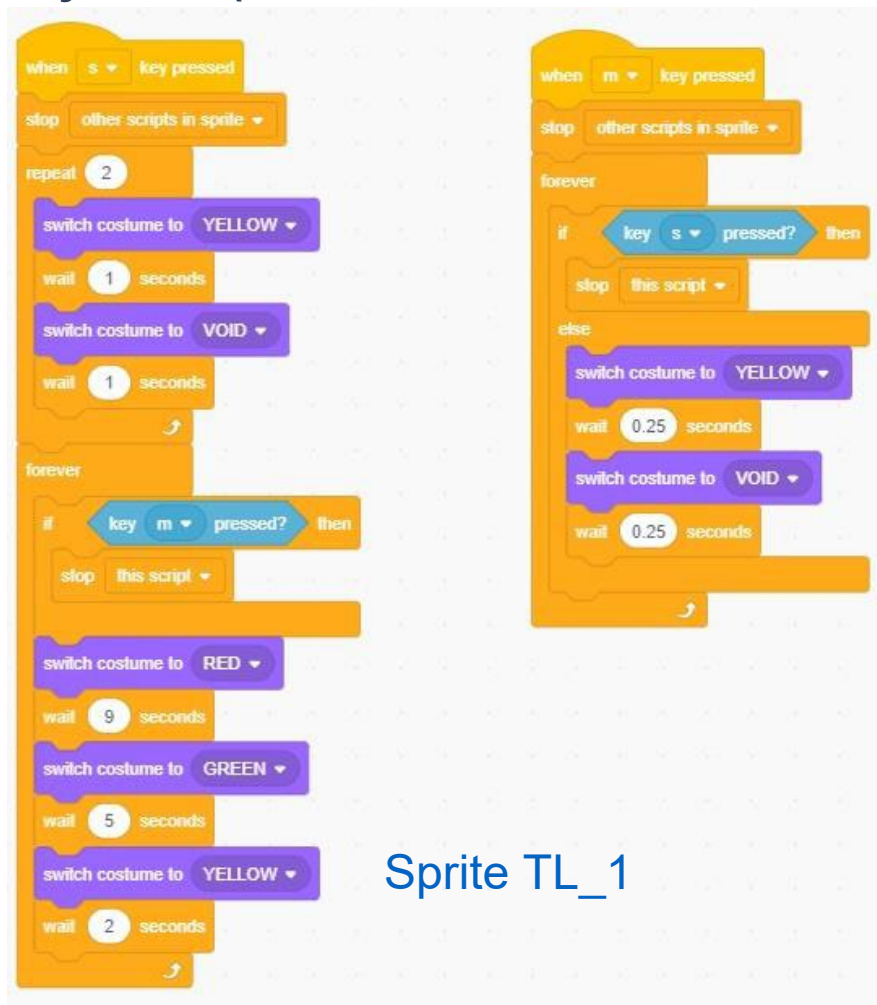
Sprite TL_1



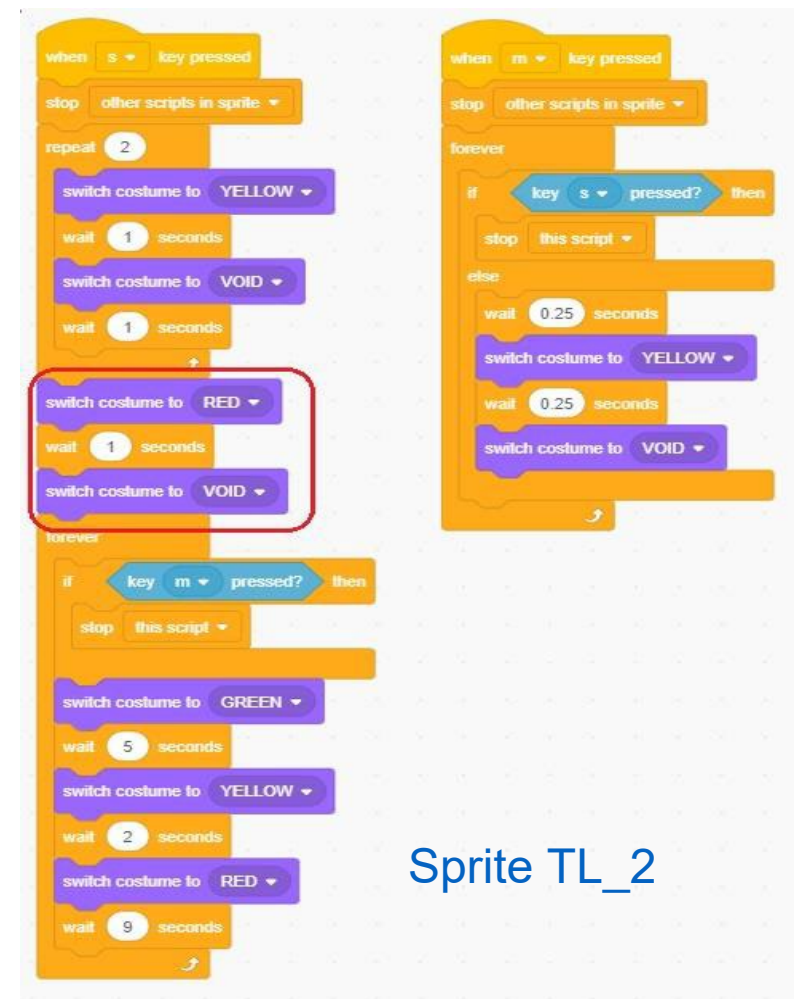
Sprite TL_2

BUILD OUR TRAFFIC LIGHTS USING SCRATCH IN “DESKTOP MODE” ON THE PC

This program controls the two traffic lights **with red light overlap** using two keys on the PC keyboard ("s" start and "m" maintenance) to activate the two scripts:



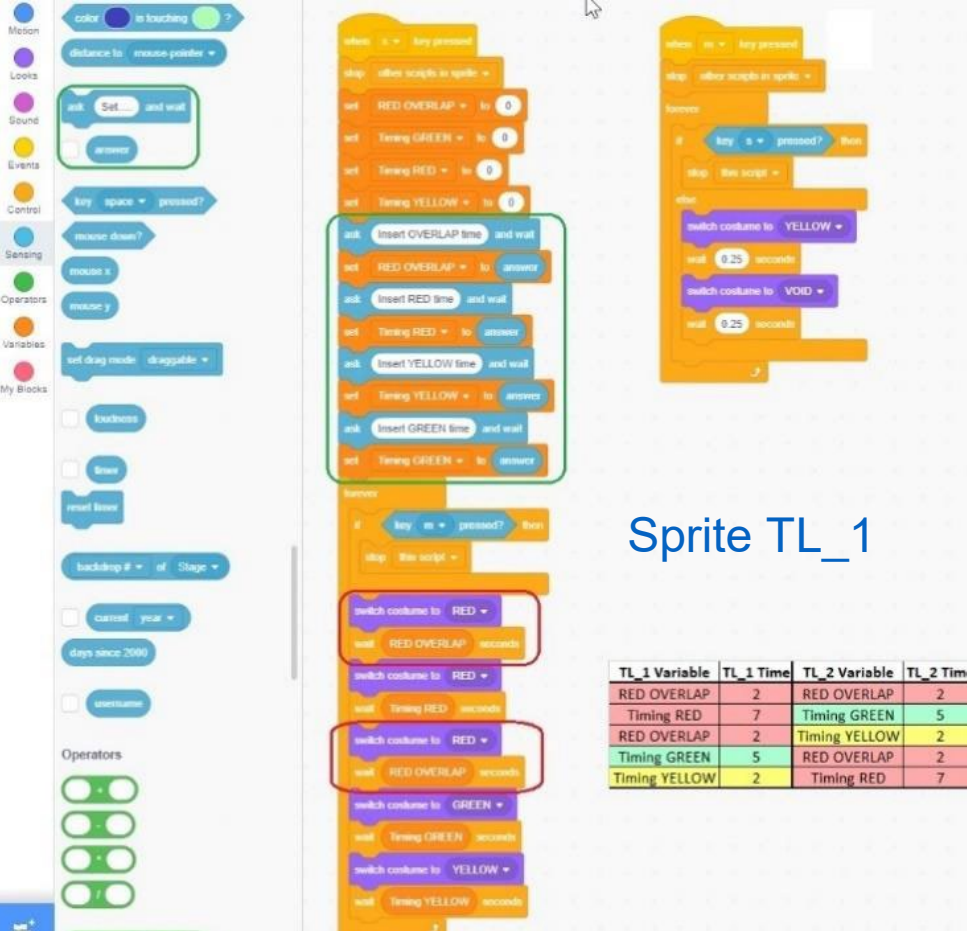
Sprite TL_1



Sprite TL_2

BUILD OUR TRAFFIC LIGHTS USING SCRATCH IN “DESKTOP MODE” ON THE PC

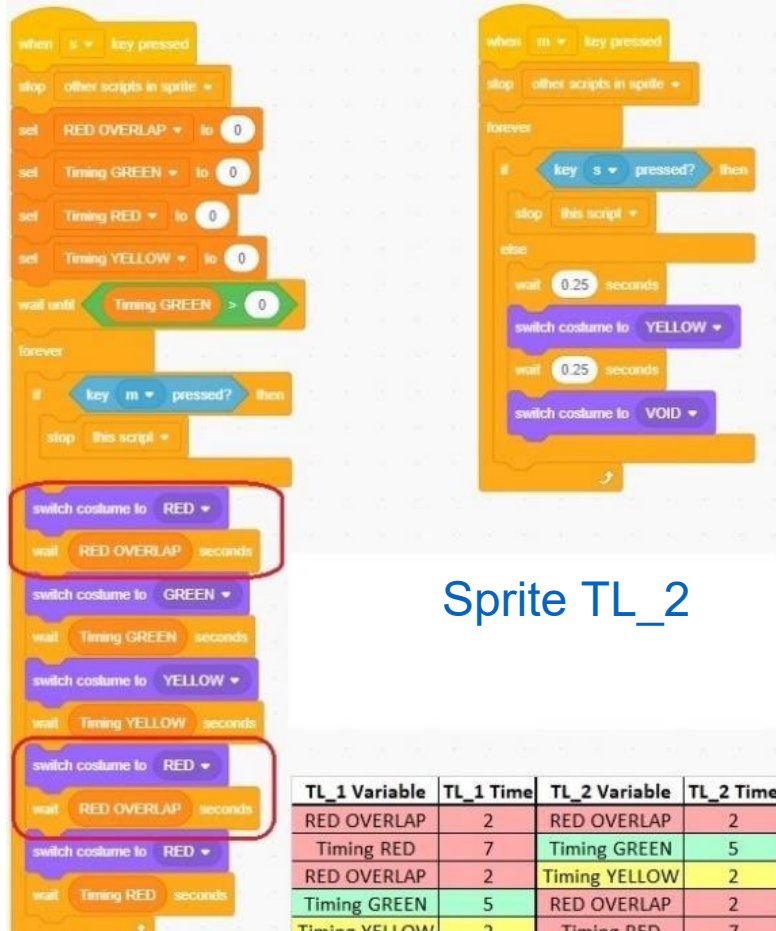
Now we introduce the possibility to define, by the operator, the duration of the individual colours using the "ask" and "answer" functions that we find under the "Sensing" group:



Scratch code for Sprite TL_1. The script starts with a 'when green flag clicked' event, followed by a 'stop all scripts in this sprite' block. It then sets four variables: RED OVERLAP to 0, Timing GREEN to 0, Timing RED to 0, and Timing YELLOW to 0. A 'forever' loop follows, containing an 'if key m pressed?' condition. If true, it switches the costume to RED, waits for RED OVERLAP seconds, switches to RED, waits for Timing RED seconds, switches to RED, waits for RED OVERLAP seconds, switches to GREEN, waits for Timing GREEN seconds, switches to YELLOW, and waits for Timing YELLOW seconds. A table below the code lists the timing values for each color.

Sprite TL_1

TL_1 Variable	TL_1 Time	TL_2 Variable	TL_2 Time
RED OVERLAP	2	RED OVERLAP	2
Timing RED	7	Timing GREEN	5
RED OVERLAP	2	Timing YELLOW	2
Timing GREEN	5	RED OVERLAP	2
Timing YELLOW	2	Timing RED	7



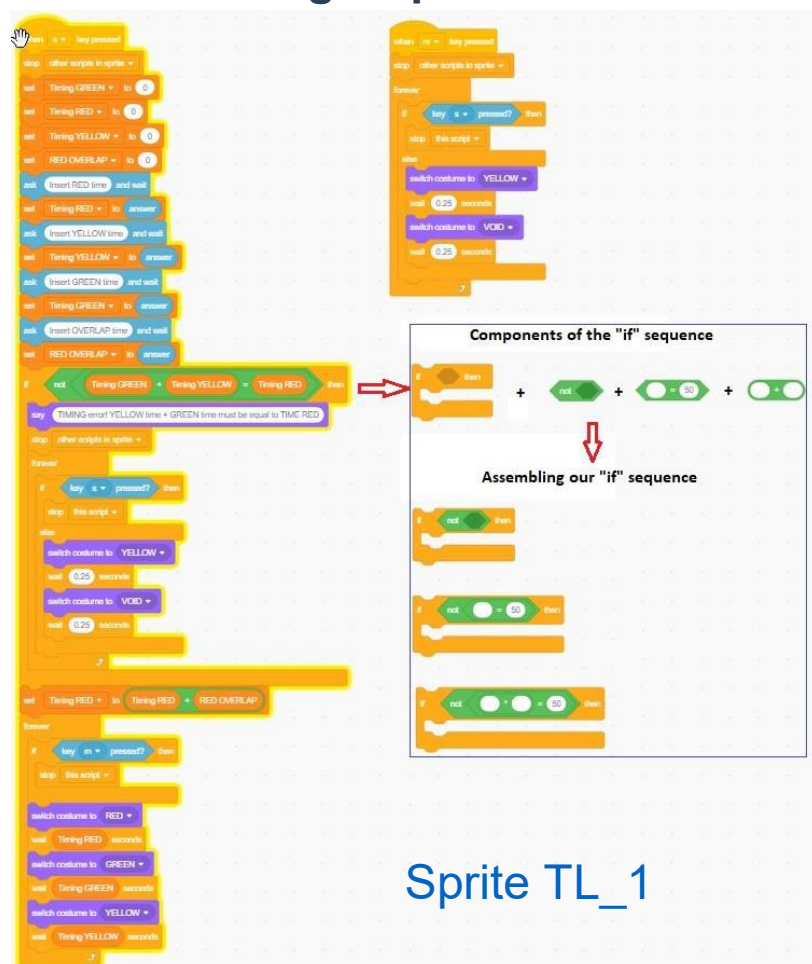
Scratch code for Sprite TL_2. The script starts with a 'when green flag clicked' event, followed by a 'stop all scripts in this sprite' block. It then sets four variables: RED OVERLAP to 0, Timing GREEN to 0, Timing RED to 0, and Timing YELLOW to 0. A 'forever' loop follows, containing an 'if key s pressed?' condition. If true, it switches the costume to YELLOW, waits for 0.25 seconds, switches to VOID, and waits for 0.25 seconds. A table below the code lists the timing values for each color.

Sprite TL_2

TL_1 Variable	TL_1 Time	TL_2 Variable	TL_2 Time
RED OVERLAP	2	RED OVERLAP	2
Timing RED	7	Timing GREEN	5
RED OVERLAP	2	Timing YELLOW	2
Timing GREEN	5	RED OVERLAP	2
Timing YELLOW	2	Timing RED	7

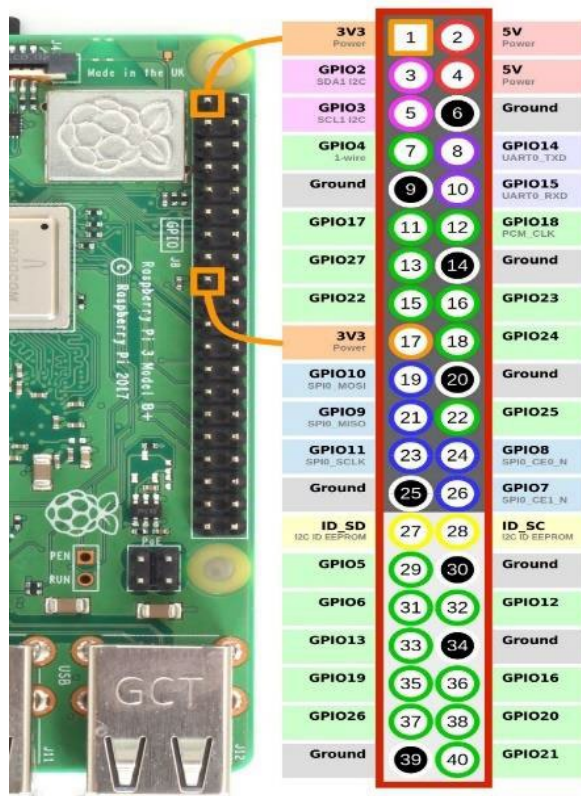
BUILD OUR TRAFFIC LIGHTS USING SCRATCH IN “DESKTOP MODE” ON THE PC

Here another improvement made introducing an “alert”, just in case if the sum of the Green+Yellow times is not equal to the Red time. This is achieved using the "say" function in the "Looks" group.



BUILD OUR TRAFFIC LIGHTS PROGRAM USING EXTERNAL DEVICES AND GPIOs

The next step is to connect the Raspberry Pi GPIOs to the breadboards. To do this it is necessary to use the jumper wires, included in the STEAM2GO Mobile Maker Space.



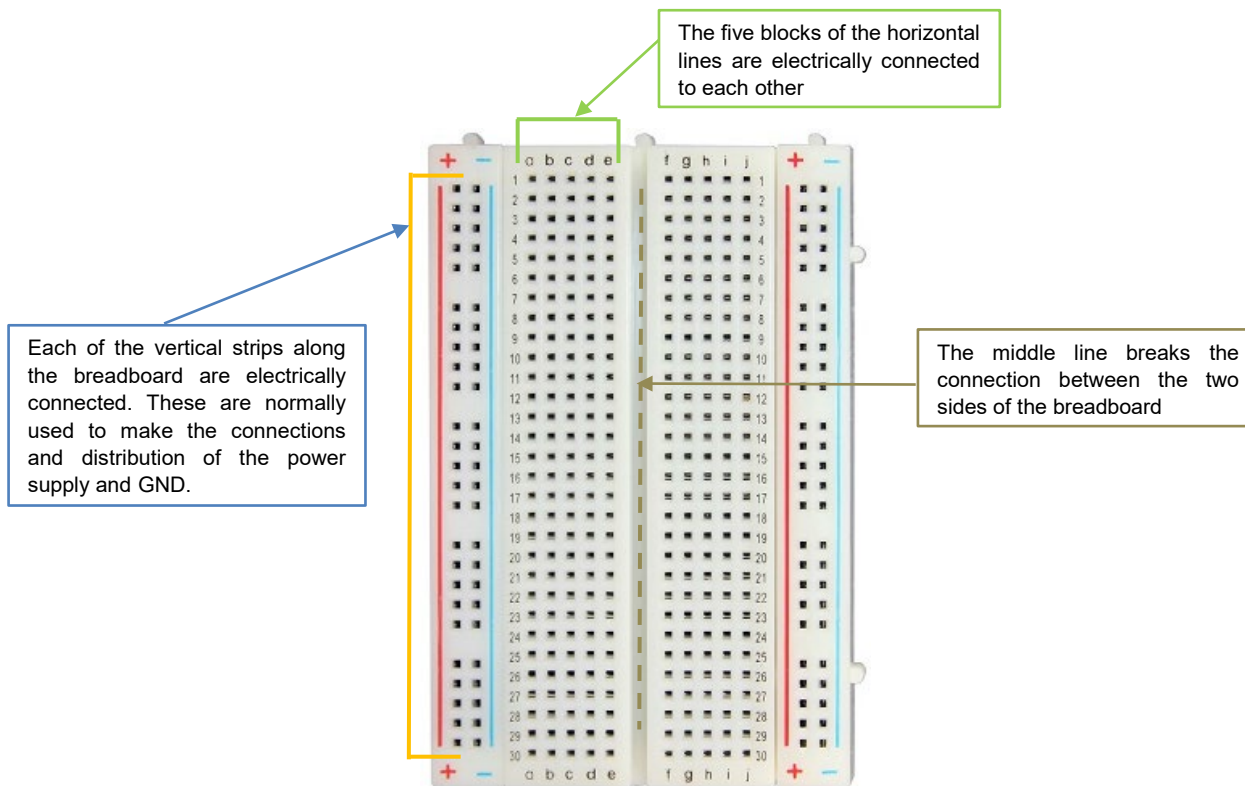
Raspberry Pi PinOut

Raspberry Pi		Led Traffic Lights		
GPIO	PIN	Traffic Light n.	Colour	Pin
GPIO10	19	1	Green 1	G
GPIO9	21	1	Yellow 1	Y
GPIO11	23	1	Red 1	R
Ground	25	1	---	GND
GPIO2	3	2	Green 2	G
GPIO3	5	2	Yellow 2	Y
GPIO4	7	2	Red 2	R
Ground	9	2	---	GND

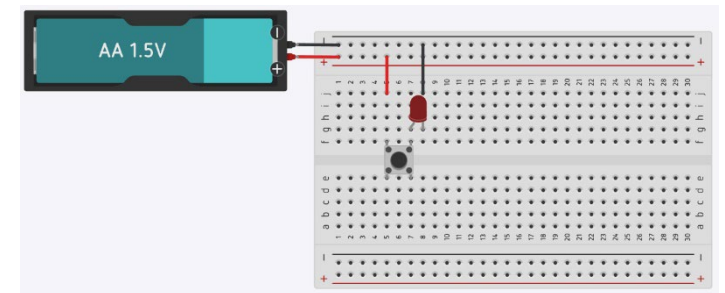
Raspberry Pi		Input devices
GPIO	PIN	Type
GPIO17	11	Start Push-button
GPIO27	13	Maintenance switch
3V3	1	+Vcc

BUILD OUR TRAFFIC LIGHTS PROGRAM USING EXTERNAL DEVICES AND GPIOs

The Breadboard is a way of constructing electronics without having to use a soldering iron. Components are pushed into the sockets (holes) on the breadboard and then extra 'jumper' wires are used to make connections between the various components



Below an example of how to connect a LED with a battery and a button



BUILD OUR TRAFFIC LIGHTS PROGRAM USING EXTERNAL DEVICES AND GPIOs

Here the first simple program that controls the two traffic lights with **NO** red light overlap using the Raspberry Pi GPIOs

Initialize GPIO 17 ("Start" button) and GPIO 27 ("Maintenance" switch) as "Pull down"

Initializes the 3 GPIOs (RED-YELLOW-GREEN) of Traffic Light 1 to "0" (turns off all 3 LEDs of TL_1)

At the beginning of the program the yellow LED blinks twice turning it on and off for 1 second

Until the GPIO27 (maintenance switch) goes to level "1" (switch activated) it repeats the RED-YELLOW-GREEN sequence



Only when the GPIO27 (maintenance) goes to level 1 (switch activate) is the "maintenance" function activated and all the other sequences of the program are stopped. The 3 LEDs are placed in the off state (turned off)

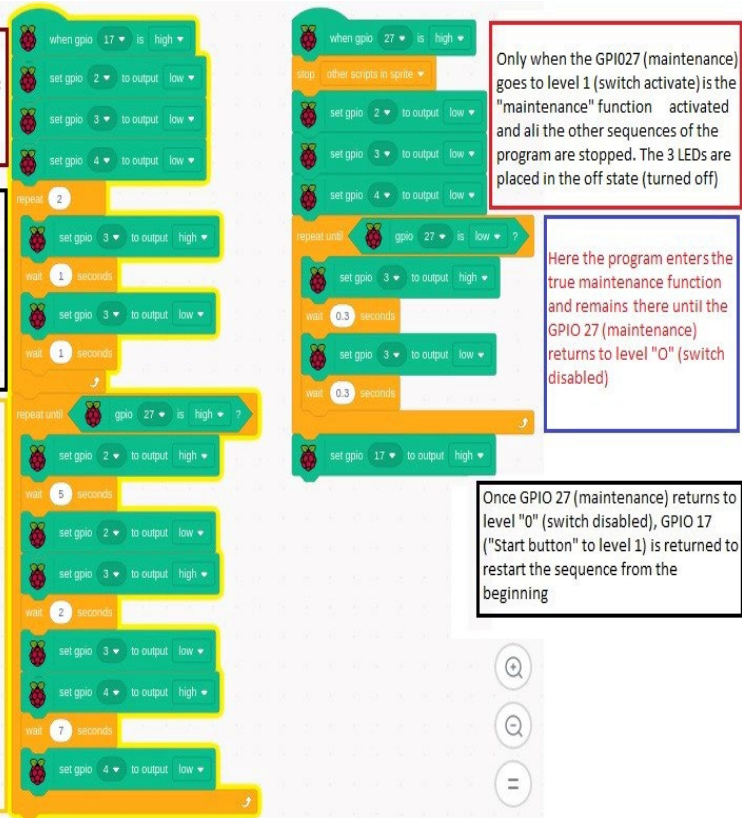
Here the program enters the true maintenance function and remains there until the GPIO 27 (maintenance) returns to level "0" (switch off)

Once GPIO 27 (maintenance) returns to level "0" (switch disabled), GPIO 17 ("Start button" to level 1) is returned to restart the sequence from the beginning

Initialize the 3 GPIOs (RED-YELLOW-GREEN) of Traffic Light 2 to "0" (turns off all 3 LEDs of TL_2)

At the beginning of the program the yellow LED blinks twice turning it on and off for 1 second

Until the GPIO 27 (maintenance switch) goes to level "1" (switch activated) it repeats the RED-YELLOW-GREEN sequence.



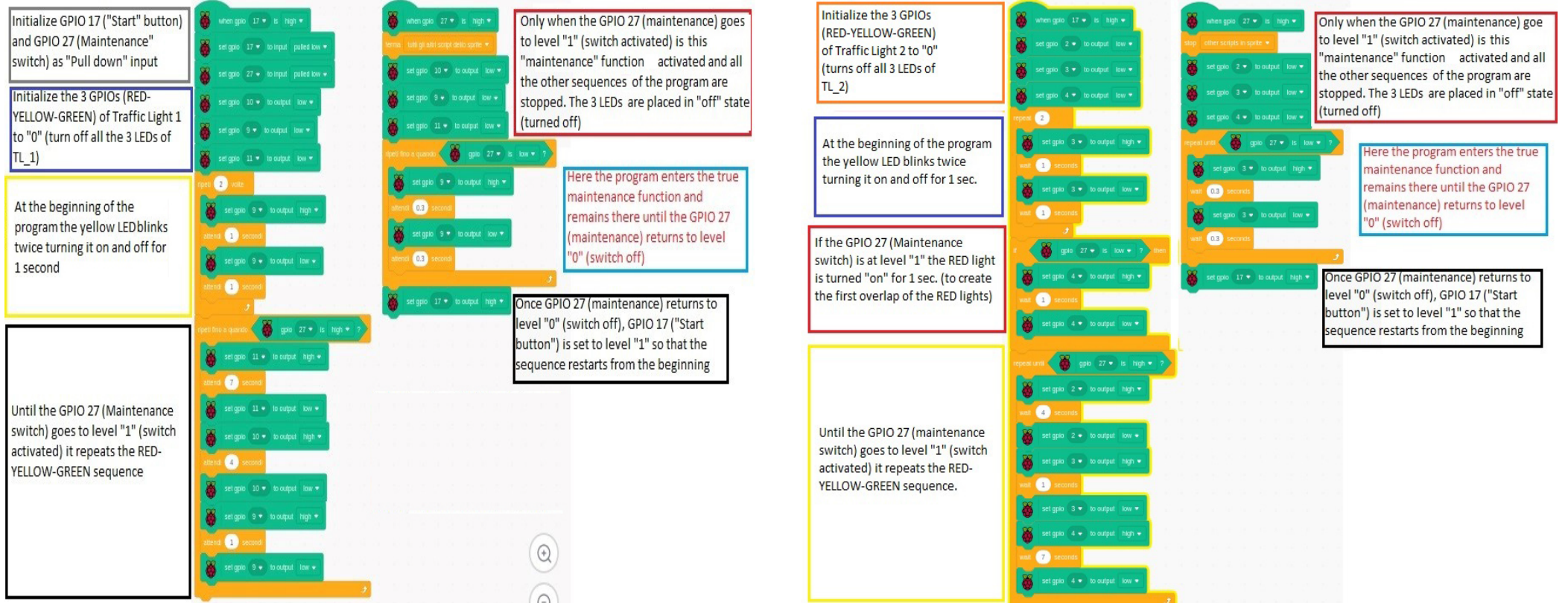
Only when the GPIO27 (maintenance) goes to level 1 (switch activate) is the "maintenance" function activated and all the other sequences of the program are stopped. The 3 LEDs are placed in the off state (turned off)

Here the program enters the true maintenance function and remains there until the GPIO 27 (maintenance) returns to level "0" (switch disabled)

Once GPIO 27 (maintenance) returns to level "0" (switch disabled), GPIO 17 ("Start button" to level 1) is returned to restart the sequence from the beginning

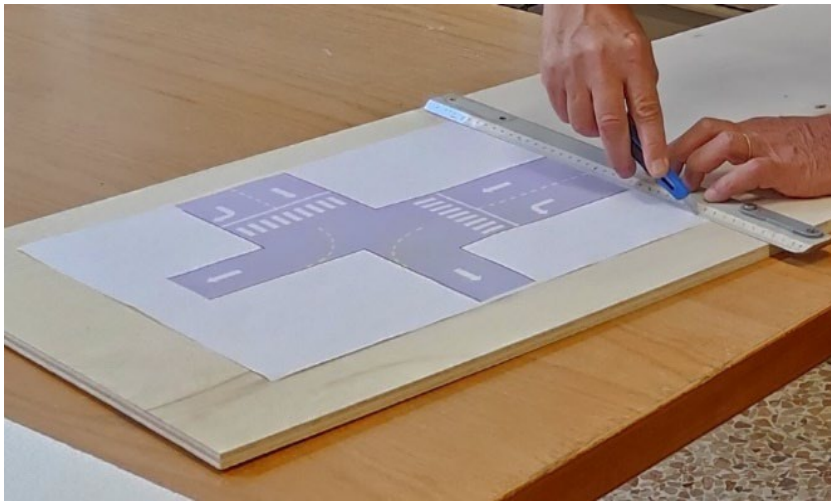
BUILD OUR TRAFFIC LIGHTS PROGRAM USING EXTERNAL DEVICES AND GPIOs

This is a bit complex program that controls the two traffic lights with red light overlap using the Raspberry Pi GPIOs

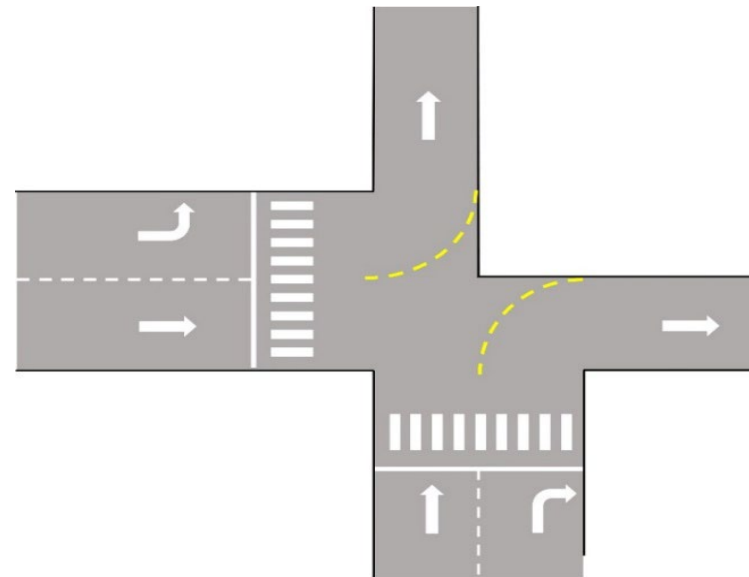


DRAWING AND CRAFTING THE FLOOR PLAN OF TWO WAYS JUNCTION

The plan of the traffic light intersection can be made with a sheet of paper of the size, as in our case, 28x34 cm., cutting out an A3 sheet or using a simple A4 sheet



The plan could be created by hand with felt-tip pens or colored pencils or by using a drawing program and printing it; in our case we used MS Word as shown below

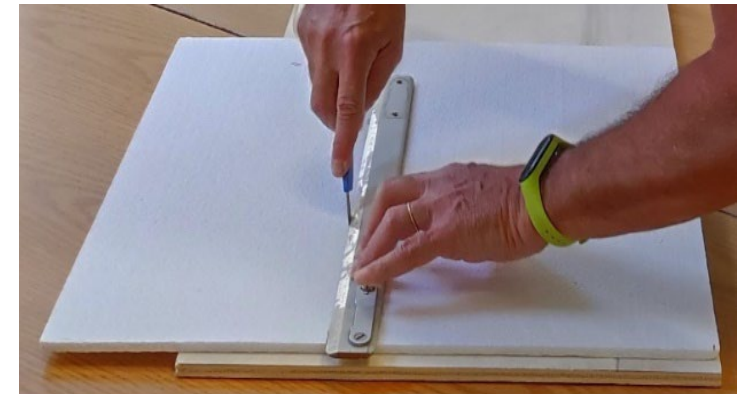


DRAWING AND CRAFTING THE FLOOR PLAN OF TWO WAYS JUNCTION

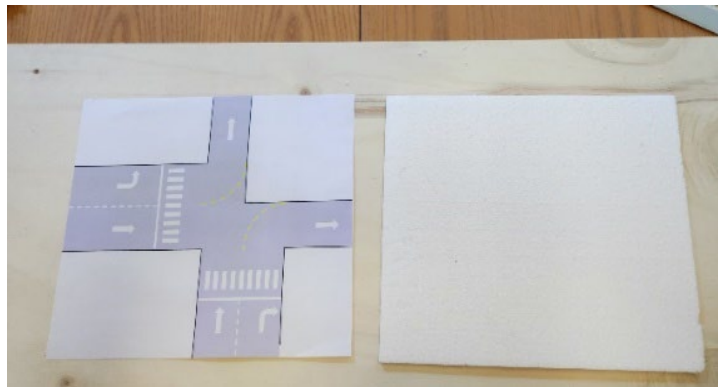
After creating the paper layout it will be necessary to create a 1 cm high styrofoam base where the breadboards will be inserted



The styrofoam sheet must have the size of the paper sheet (you can use a A4 sheet) used to create the graphic layout

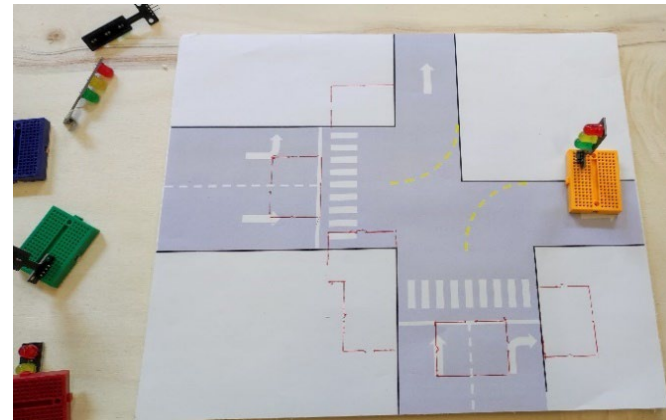
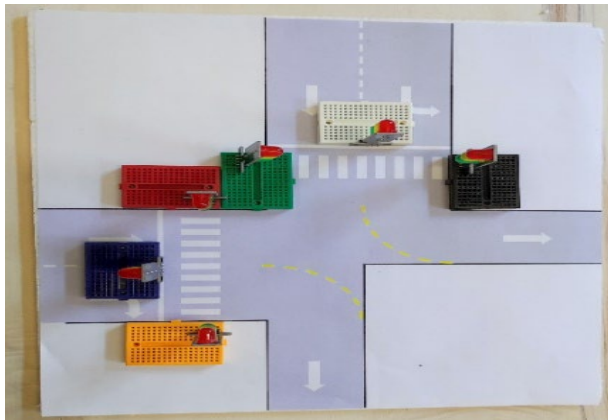


The layout and the styrofoam sheets cut to size are shown below

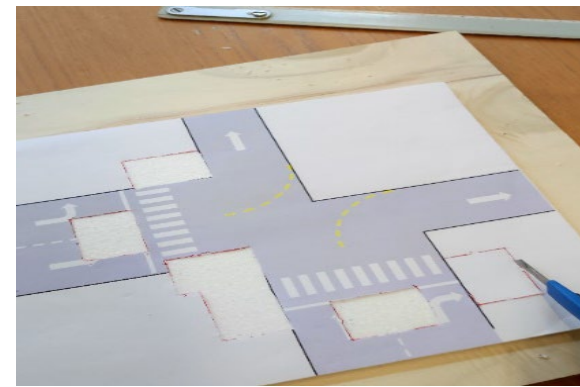
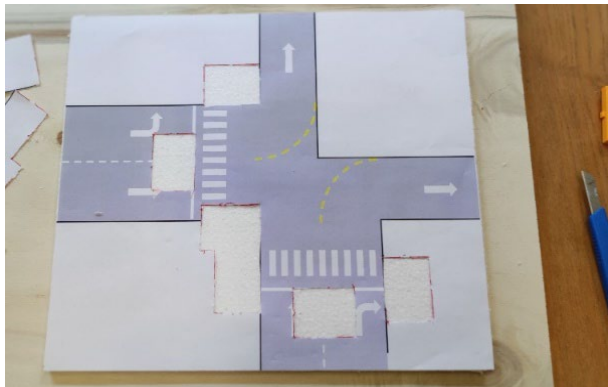


DRAWING AND CRAFTING THE FLOOR PLAN OF TWO WAYS JUNCTION

The next step will be to mark the exact position of the breadboards on a drawing of the intersection using a previously created layout

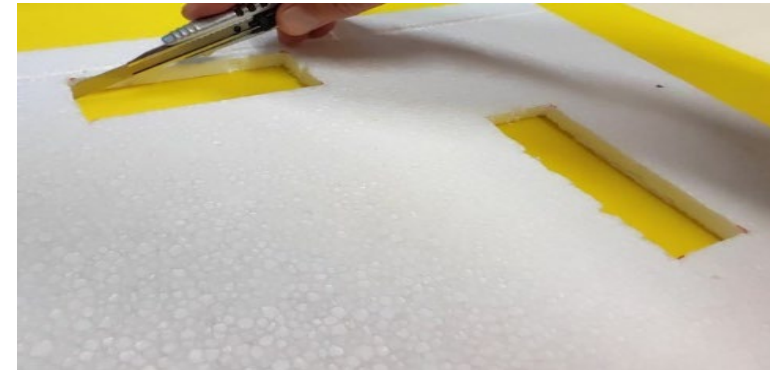
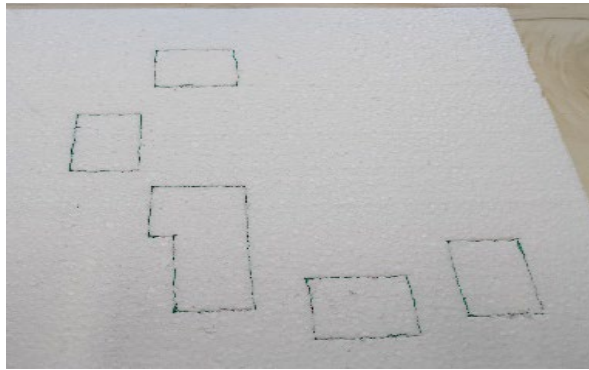
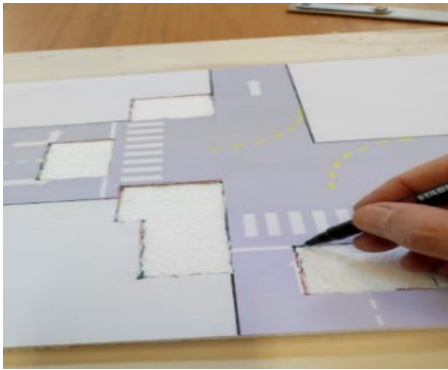


Now we have to cut out the breadboard slots in the styrofoam sheet to insert the breadboards in

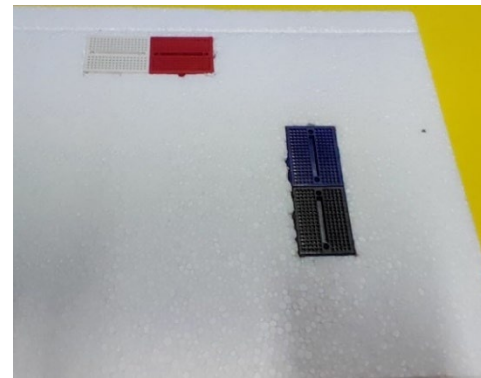
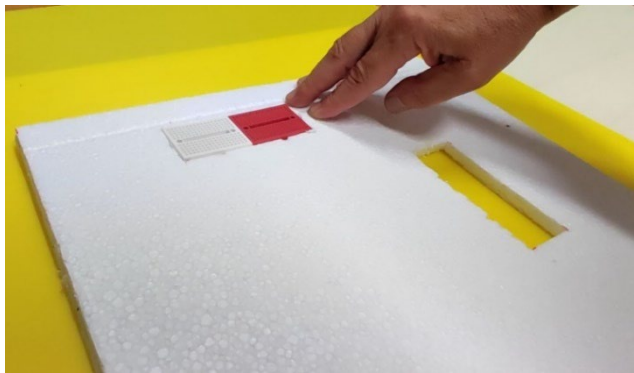


DRAWING AND CRAFTING THE FLOOR PLAN OF TWO WAYS JUNCTION

Now we will cut out the breadboard slots in the styrofoam sheet to insert the breadboards in

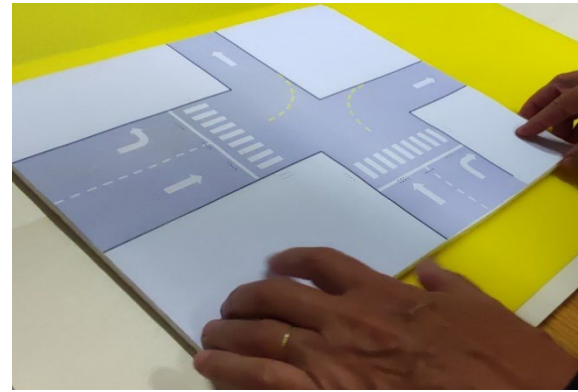
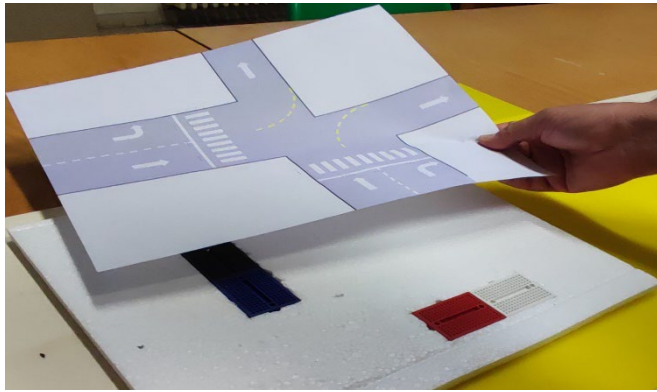


Now we have to cut out the breadboard slots in the styrofoam sheet to insert the breadboards in

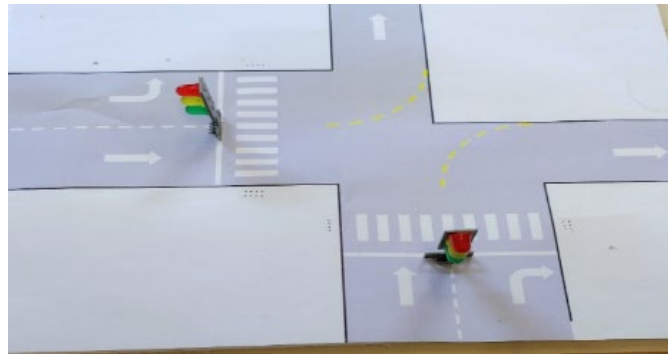


DRAWING AND CRAFTING THE FLOOR PLAN OF TWO WAYS JUNCTION

We can now put the plan on a alveolar polypropylene substrate and proceed to insert the mini led traffic lights on the relative breadboard

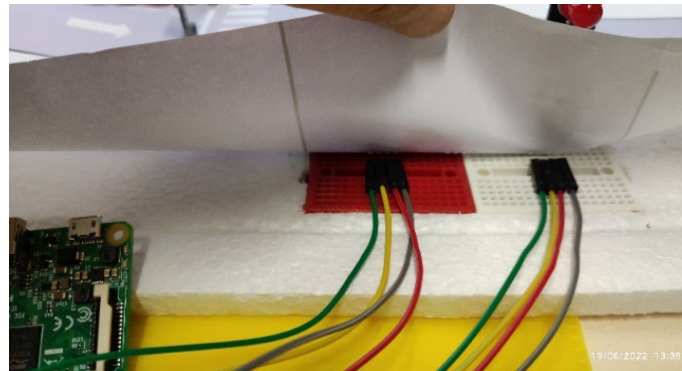


Below the layout of the complete traffic light with the two LEDs traffic lights inserted in the breadboard

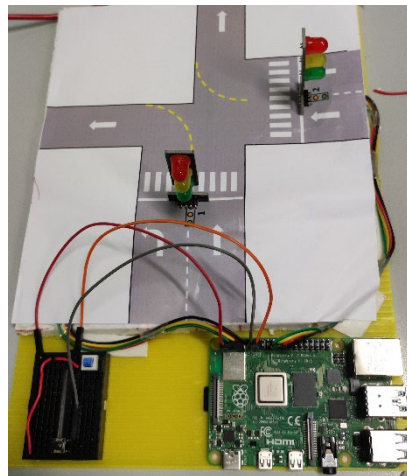


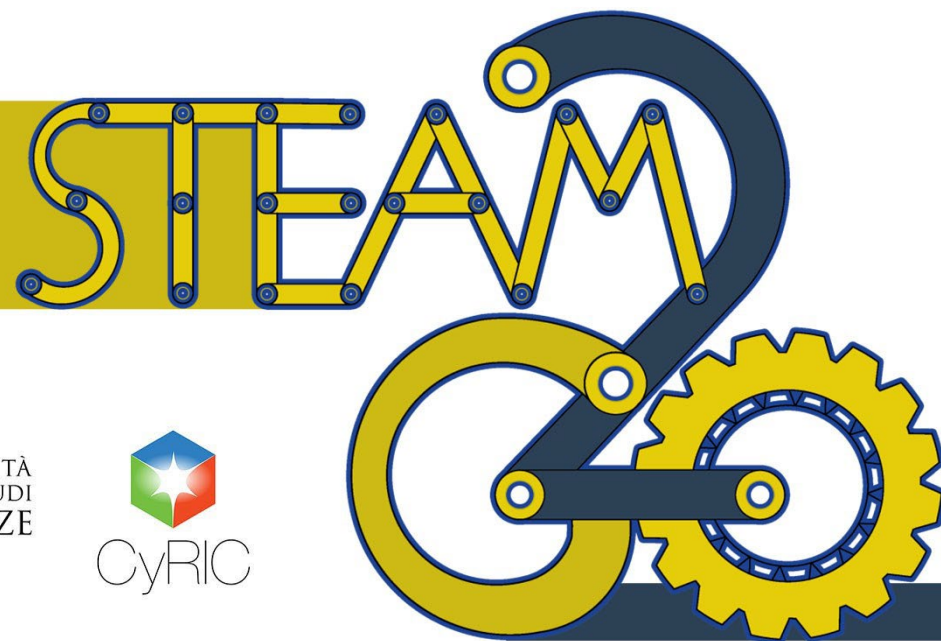
DRAWING AND CRAFTING THE FLOOR PLAN OF TWO WAYS JUNCTION

Last step: connect the LED devices to the GPIO of the Raspberry Pi



Here an example of the complete system





Erasmus+

