

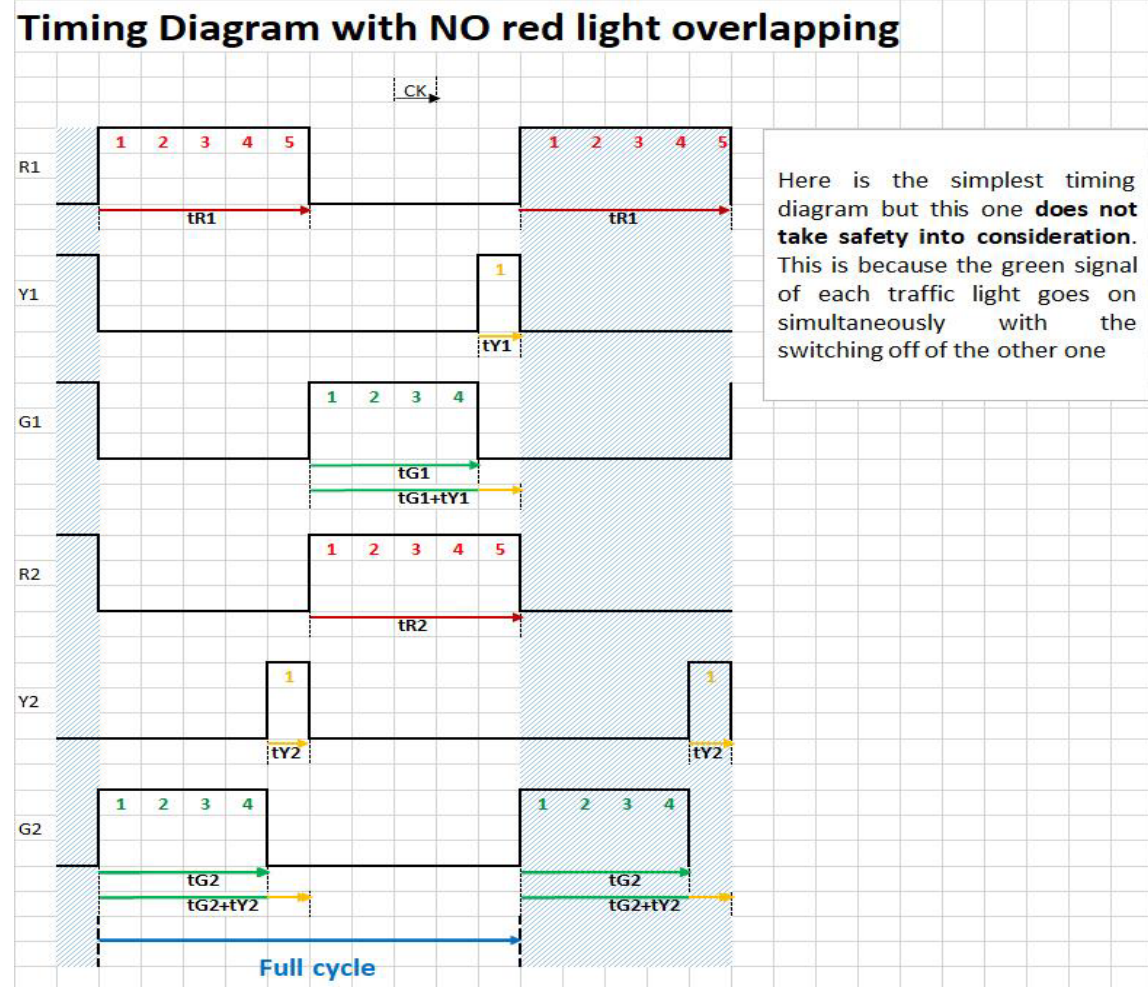
Renato Gatti – Loris Ranalli



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

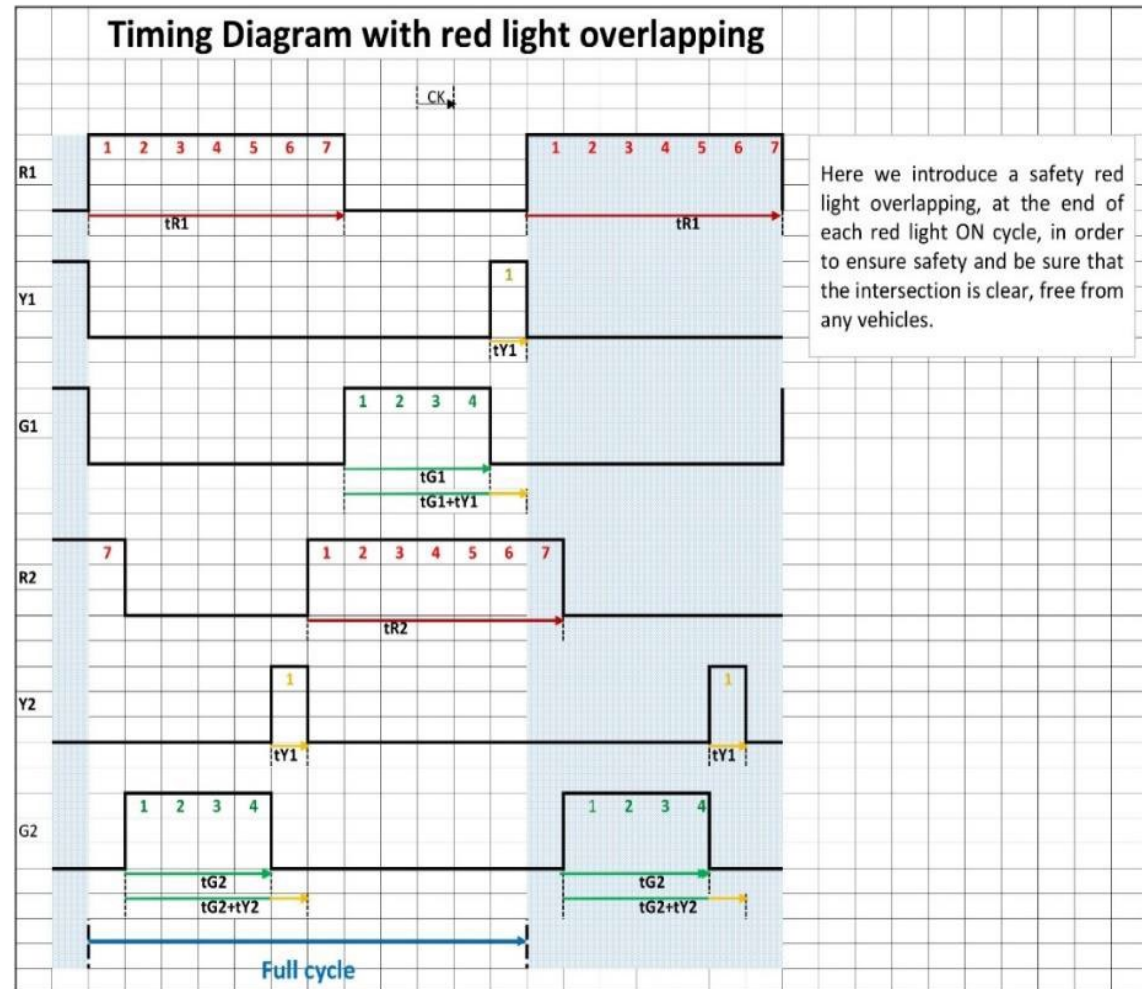
ABOUT THE PROJECT: REALIZATION OF THE TIMING DIAGRAMS

Simplest Timing Diagram with NO red light overlap



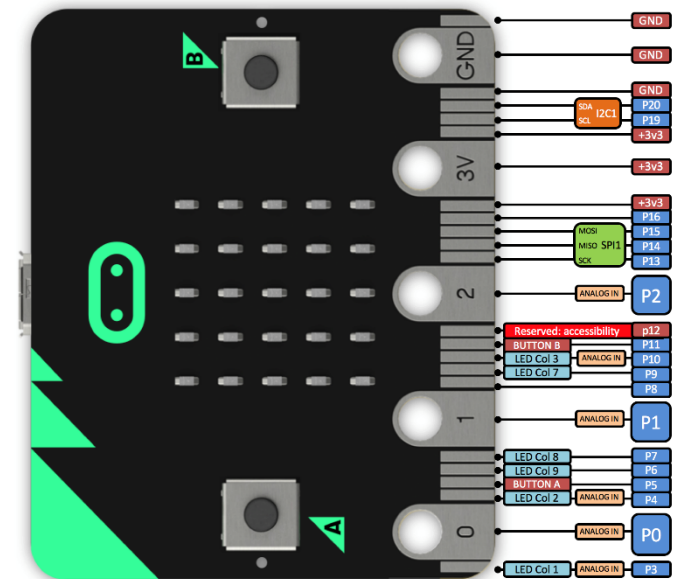
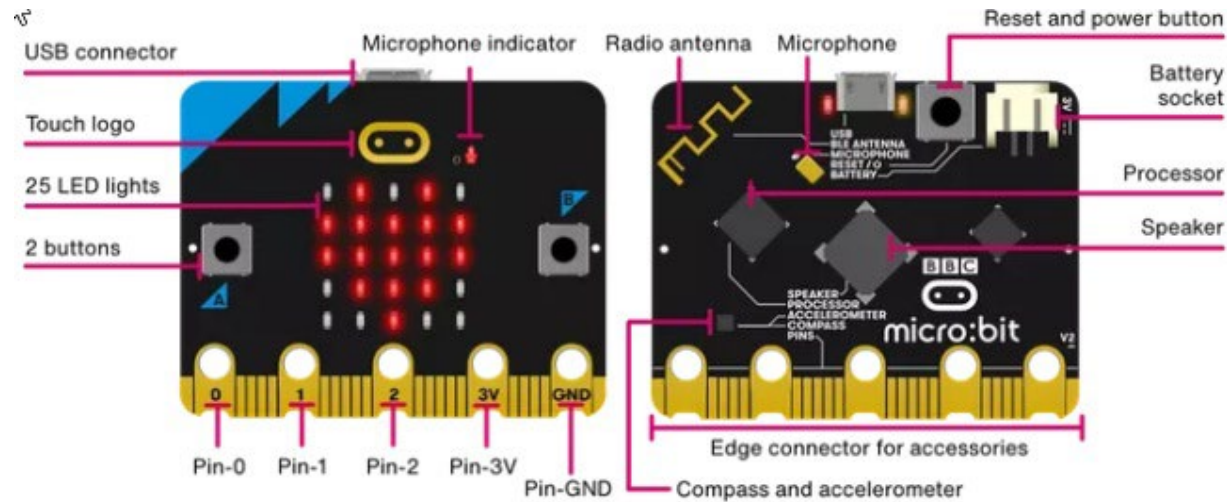
ABOUT THE PROJECT: REALIZATION OF THE TIMING DIAGRAMS

Timing diagram with red lights overlap



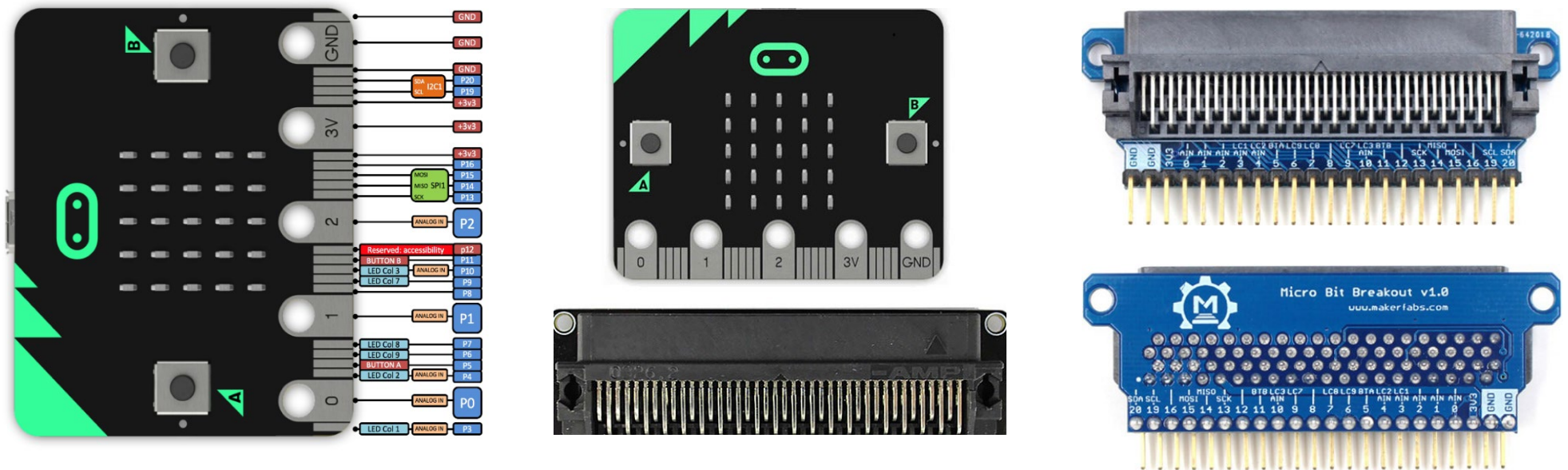
OUR "TRAFFIC LIGHTS" USING MICRO:BIT and MAKECODE

Introducing the Micro:Bit board: the layout and the “standard” pinout of the Micro:Bit board



OUR "TRAFFIC LIGHTS" USING MICRO:BIT and MAKECODE

Introducing the Micro:Bit board: the “advanced” pinout of the Micro:Bit board

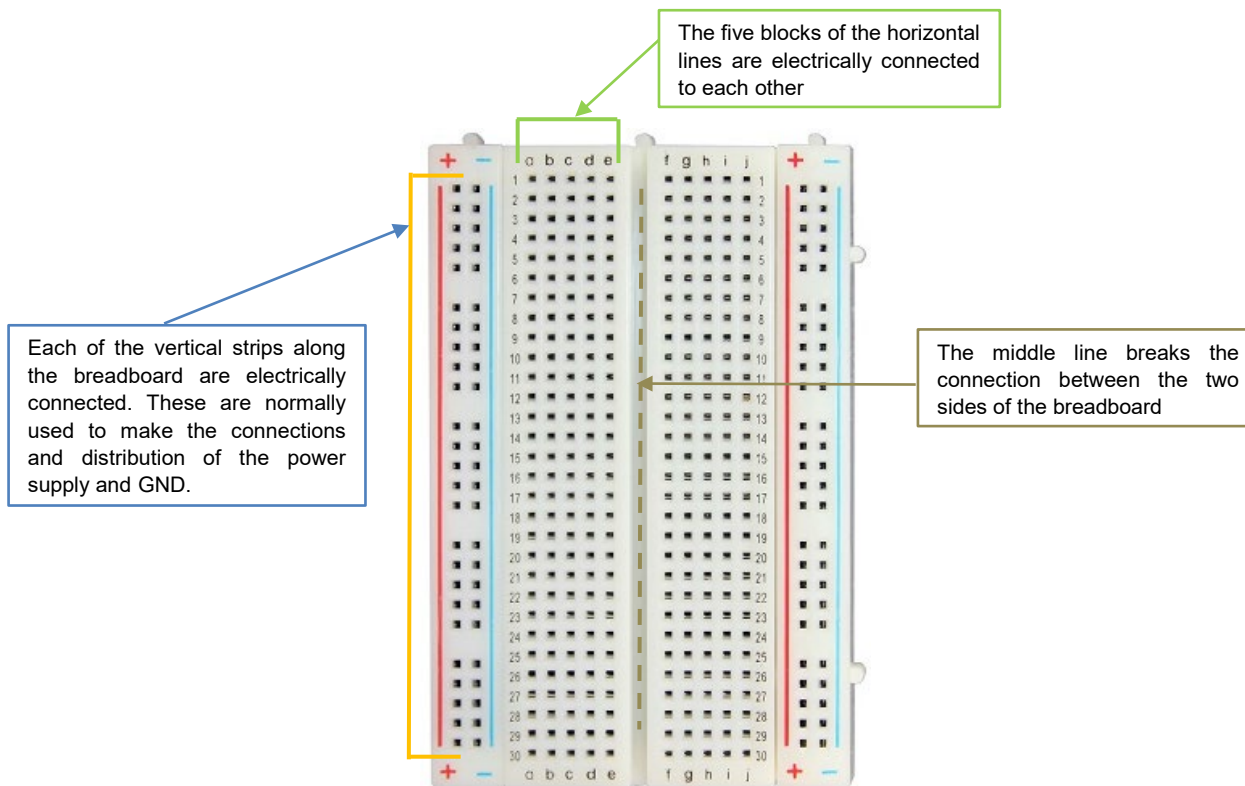


If we want to use all the I/O pins of the Micro:Bit board, as in our case, it's necessary to add the breakout board for Micro:Bit and the jumper wires, included in the Mobile Maker Space.

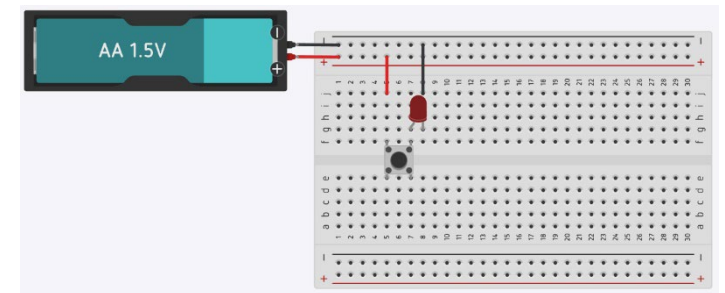
The Micro:Bit breakout board I/Os are shown in the figure above.

BUILD OUR TRAFFIC LIGHTS PROGRAM USING THE BREAD BOARD

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 THE MICRO:BIT I/Os

It's convenient to make a spreadsheet with the interconnections between the Micro:Bit I/Os and the traffic lights on the breadboards, as shown below:

Micro:bit		Led Traffic Lights	
PIN	Traffic Light n.	Colour	Pin
16	1	Green 1	G
15	1	Yellow 1	Y
14	1	Red 1	R
GND	1	---	GND
13	2	Green 2	G
9	2	Yellow 2	Y
8	2	Red 2	R
14	2	---	GND

Micro:bit	Input device
PIN	Type
0	Start Push-button
2	Maintenance switch
3V3	+Vcc

OUR "TRAFFIC LIGHTS" USING MICRO:BIT and MAKECODE

Programming the Micro:Bit board using MakeCode software

It is possible to program the Micro:bit board through an online application, MakeCode, provided by Microsoft, which we find here:

<https://makecode.microbit.org>

or you can download an offline application here:

<https://makecode.microbit.org/offline-app>

The offline app works the same as the Web application does in your browser but it's a stand-alone application that will work when a connection to the internet is restricted or not available.

OUR "TRAFFIC LIGHTS" USING MICRO:BIT and MAKECODE

Programming Micro:Bit Traffic Light using internal "A" pushbutton

Hereinafter are shown the first two projects that DON'T use external buttons.

Project 1: Traffic lights system with NO red light overlap



When the "A" button is pressed, the sequence begins, making both yellow lights flash for 2 seconds.

Then the program enters the infinite working loop.

In this program, the timings, used as an example, are the following:

RED = 7 seconds

YELLOW = 2 second

GREEN = 5 seconds

OUR "TRAFFIC LIGHTS" USING MICRO:BIT and MAKECODE

Programming Micro:Bit Traffic Light using internal "A" pushbutton

Project 2: Traffic lights system with red light overlap



The program is similar to the one already done (TL-NO-overlap-NO-buttons).

When the "A" button is pressed, the sequence begins, making both yellow lights flash two times for 0.5 seconds. The only difference is that the timings of the lights will have to take into account the overlap of the red lights.

Here the sequence:

- 1) RED_1 (P14) and GREEN_2 (P13) = "ON" for 5 seconds - Then
 - 2) GREEN_2 (P13) = "OFF" and {RED_1 (P14) and YELLOW_2 (P9)} = "ON" for 2 seconds - Then
 - 3) YELLOW_2 (P9) = "OFF" and RED_2 (P8) = "ON" for 2 seconds (here the 2 seconds red lights overlap) - Then
 - 4) RED_1 (P14) = "OFF" and {RED_2 (P8) and GREEN_1 (P16)} = "ON" for 5 seconds - Then
 - 5) GREEN_1 (P16) = "OFF" and {RED_2 (P8) and YELLOW_1 (P15)} = "ON" for 2 seconds - Then
 - 6) YELLOW_1 (P15) = "OFF" and RED_1 (P14) = "ON" for 2 seconds (here the 2 seconds red lights overlap) - Then
 - 7) RED_1 (P14) and RED_2 (P8) = "OFF" -Then
- The cycle restarts from step 1)

OUR "TRAFFIC LIGHTS" USING MICRO:BIT and MAKECODE

Programming Micro:Bit Traffic Light using external "Start" and "Maintenance" buttons

Here we use two external input devices, a pushbutton to "Start" the operating sequence and a switch to enter in "Maintenance" mode. In "Maintenance" mode the yellow lights flash every 0,20 seconds (200.000 microseconds).

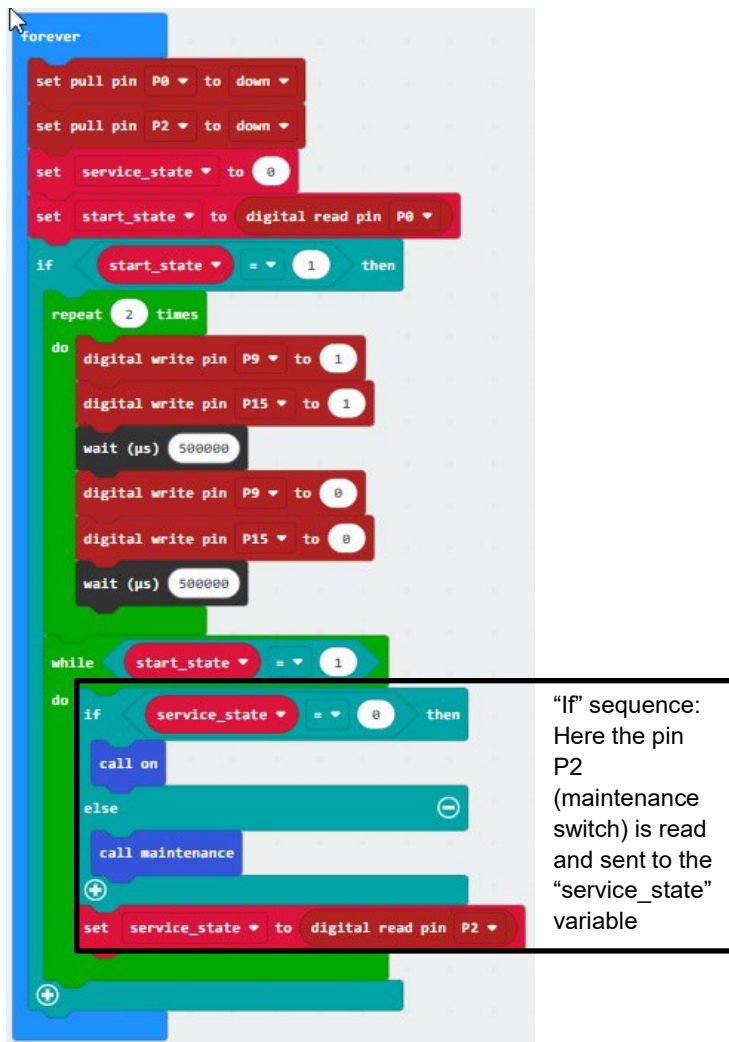
We have to create the three functions:

- 1) "forever" function that starts as soon as we turn on the device
- 2) "maintenance" function which starts the maintenance sequence (the yellow lights alternately flash every 200 milliseconds)
- 3) "on" function which starts the normal operating sequence

The first two functions, 1) "forever" and 2) "maintenance", remain unchanged whether the program is implemented with or without overlapping of the red lights.

OUR "TRAFFIC LIGHTS" USING MICRO:BIT and MAKECODE

Project 3: Traffic lights with NO red light overlap using external "Start" and "Maintenance" buttons



When the machine is powered up, the digital pins P0 (Start Push-button) and P2 (Maintenance switch) are put into pull-down mode. Then the two variables are set:

- `service_state`: is initialized to 0 (OFF)
- `start_state`: this variable is assigned the value that is read from the `digital_pin_1`, which represents the status of the P0 "Start" Push-button

So, if the P0 (Start Push-button) is pressed the sequence starts, otherwise it remains in waiting state. Once the sequence has started, the yellow lights blink twice for 0.5 seconds ("repeat/do") and then enters in the "while/do" condition. Here we found the logic block "if/else" which checks the value of the "service_state" variable, which depends on the "Maintenance_switch" status:

- if "Maintenance_switch" is OFF (0) the function "on" is called and the system enters normal operation
- else "Maintenance_switch" is ON (1) the function "maintenance" is called and the system enters the maintenance mode, making the yellow lights flash "n" times (in our case 2) before checking again the else function "Maintenance_switch"
- once the maintenance switch is turned off again, the control sequence returns the program to the "on" function

OUR "TRAFFIC LIGHTS" USING MICRO:BIT and MAKECODE

Project 4: Traffic lights with red light overlap using external “Start” and “Maintenance” buttons



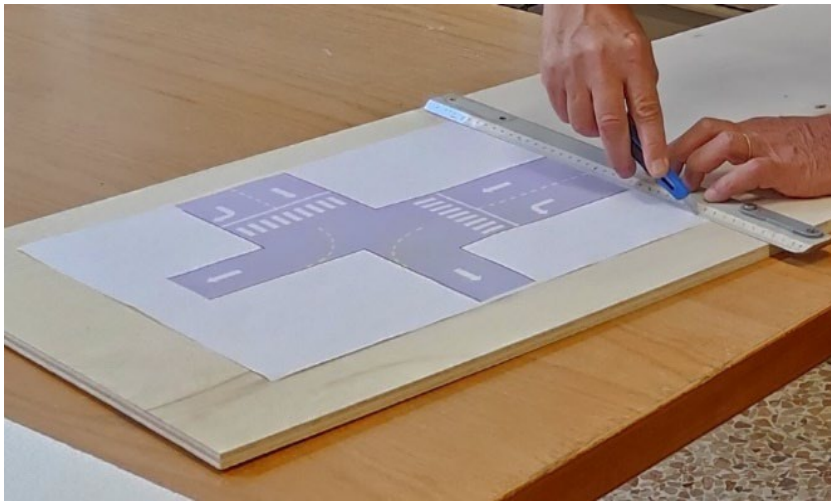
In this program the two functions, 1) “forever” and 2) “maintenance” remain unchanged whether the program is implemented with or without overlapping of the red lights. Please refer to the functions already previously explained at page 12.

The “on” function is very similar to the same as we saw in the project 3.

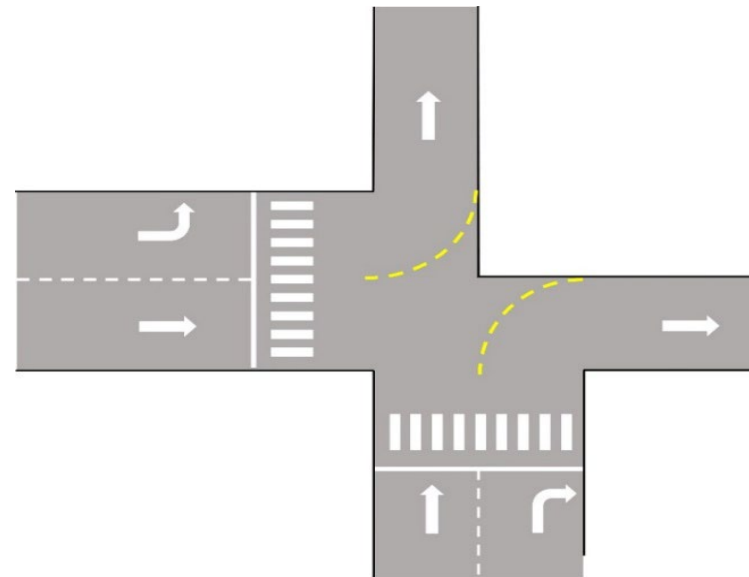
The new “on” function is shown in the figure on the left.

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

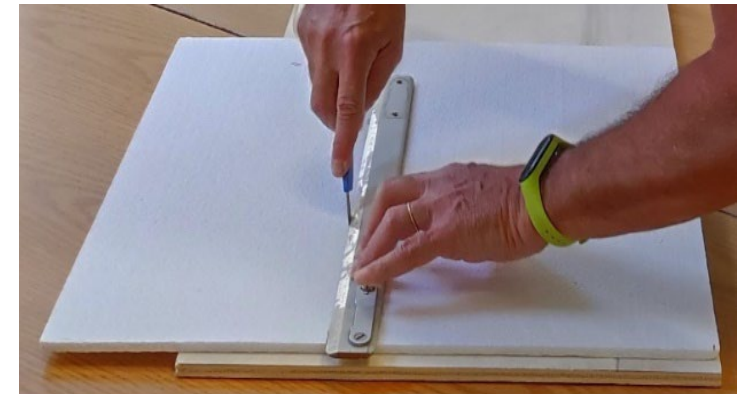


DRAWING AND CRAFTING THE FLOOR PLAN OF TWO WAYS JUNCTION

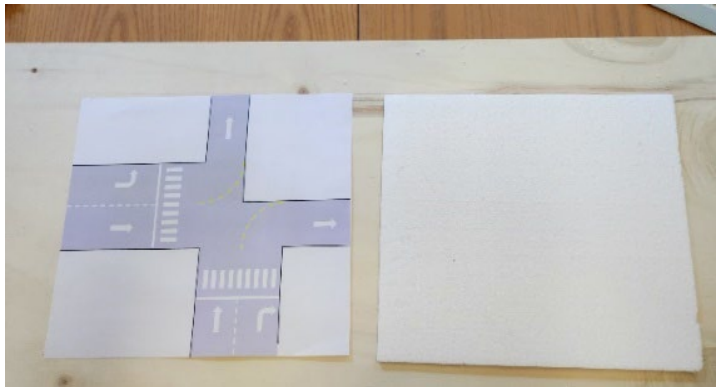
After creating the paper layout it will be necessary to create a 1 cm high styrofoam base where the breadboards will be inserted



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

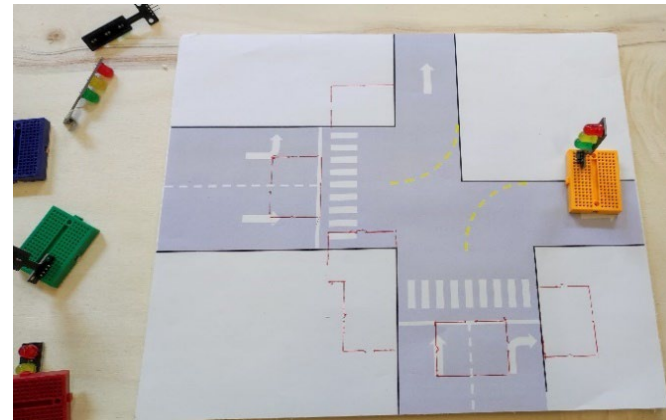
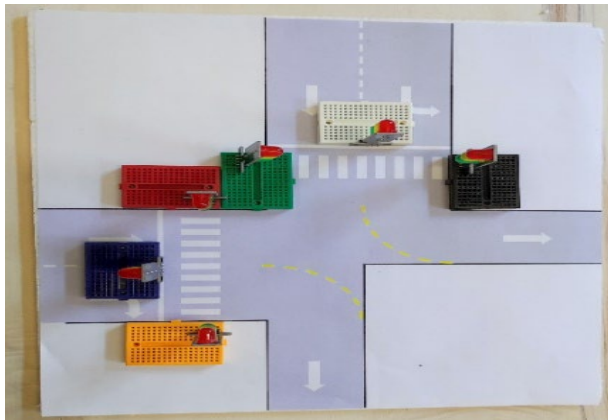


The layout and the styrofoam sheets cut to size are shown below

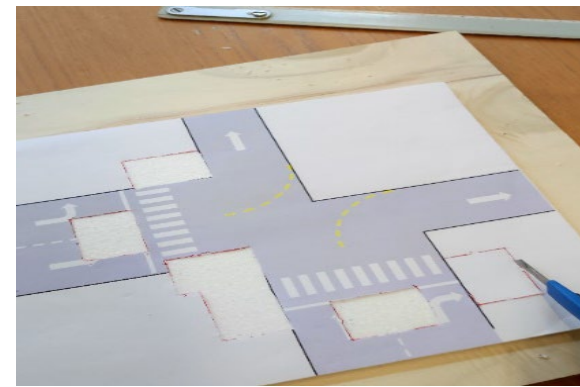
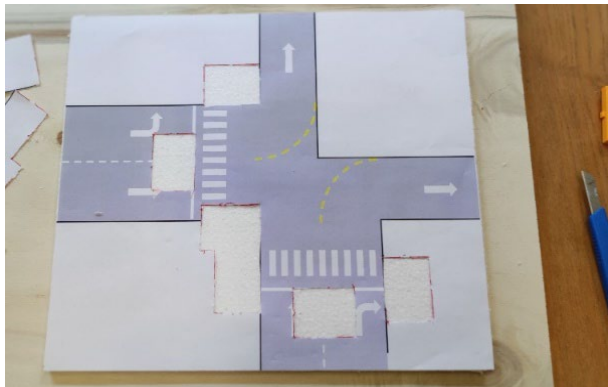


DRAWING AND CRAFTING THE FLOOR PLAN OF TWO WAYS JUNCTION

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

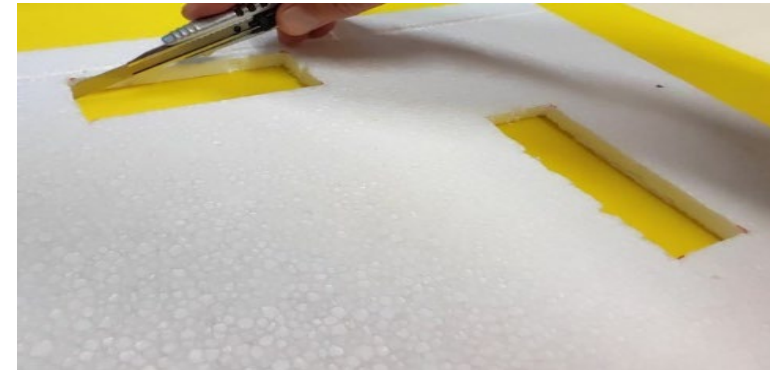
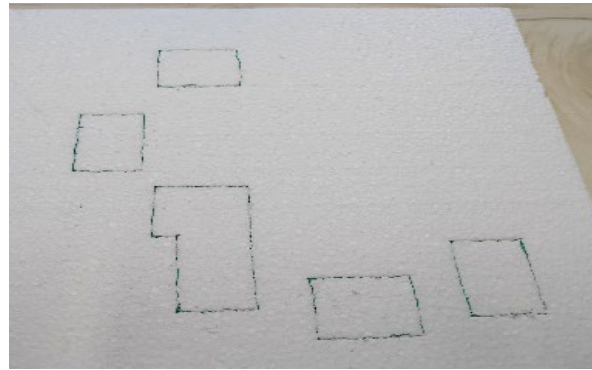
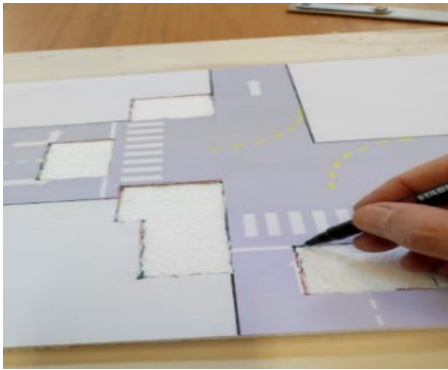


Now we have to cut out the breadboard slots in the styrofoam sheet to insert the breadboards in

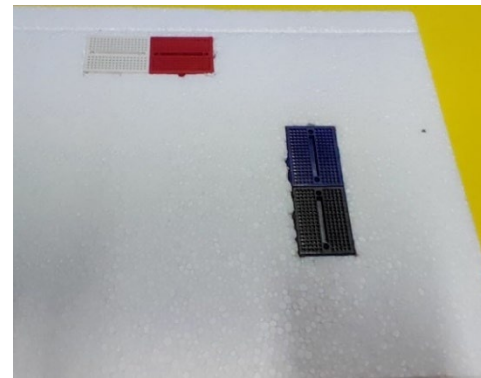
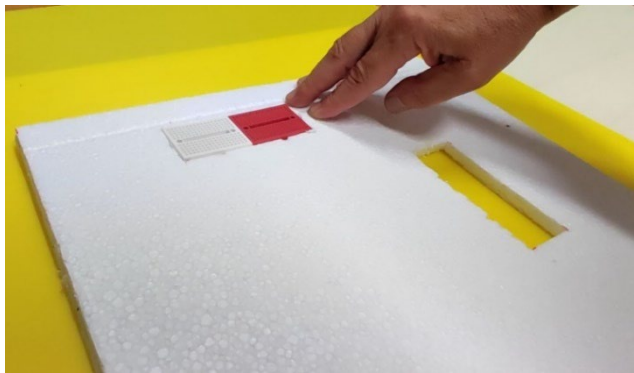


DRAWING AND CRAFTING THE FLOOR PLAN OF TWO WAYS JUNCTION

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

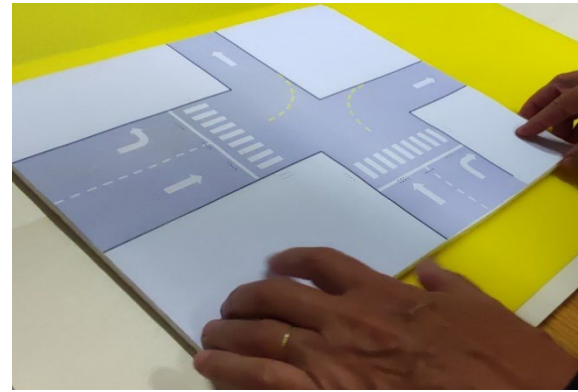
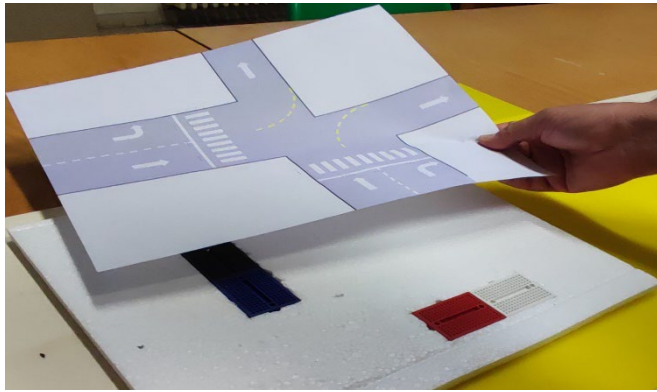


Now we have to cut out the breadboard slots in the styrofoam sheet to insert the breadboards in

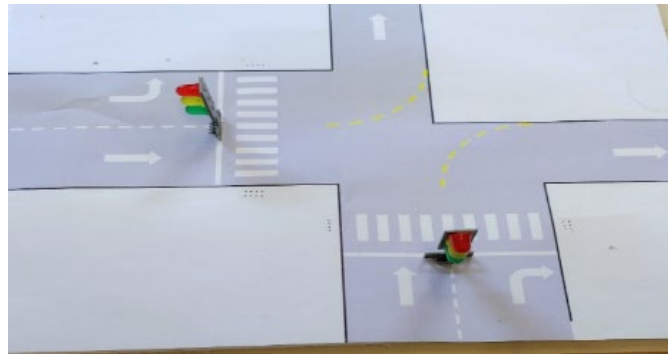


DRAWING AND CRAFTING THE FLOOR PLAN OF TWO WAYS JUNCTION

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

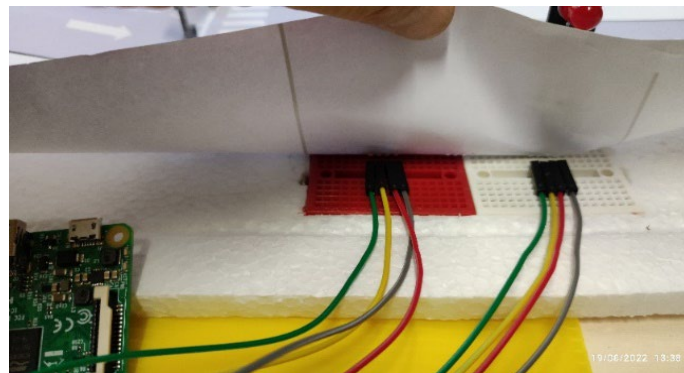


Below the layout of the complete traffic light with the two LEDs traffic lights inserted in the breadboard

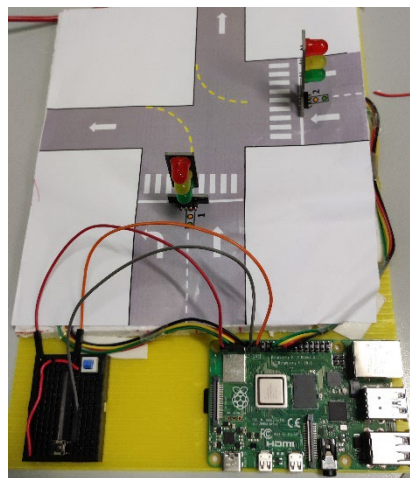


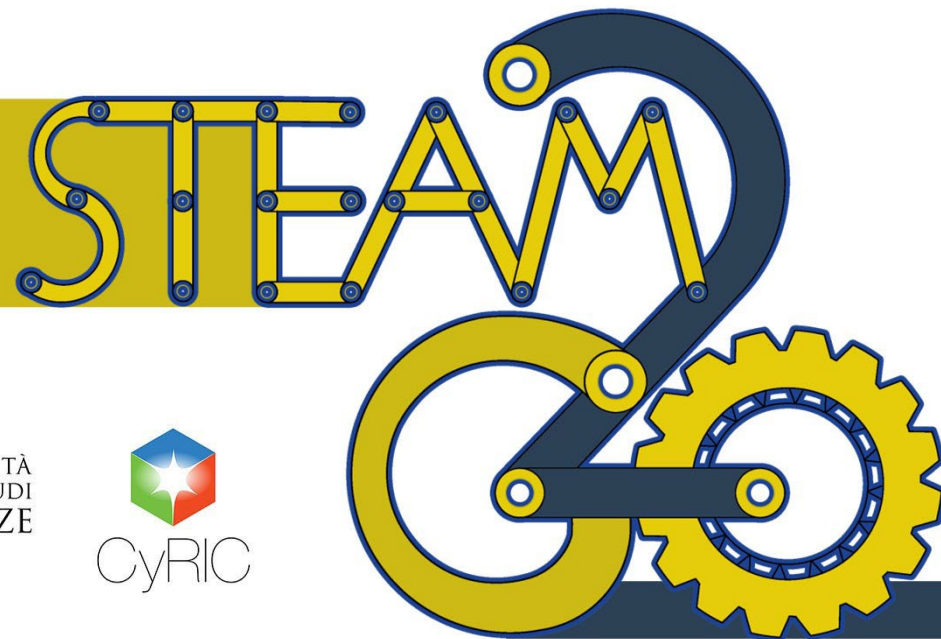
DRAWING AND CRAFTING THE FLOOR PLAN OF TWO WAYS JUNCTION

Last step: connect the LED devices to the GPIO of the Micro:Bit board



Here an example of the complete system





Erasmus+

