



0_0

0 () 000

円

0_0

0 () 000

찌찌

0 () 000

STEMJAM Teaching Guide

Developing make spaces to promote creativity around STEM in schools Acronym: STEMJAM Project no. 2016-1-ES01-KA201-025470

www.stemjam.eu

0 () 000

出出

0 0 000

0 () 000

Co-funded by the _____ Erasmus+ Programme 5 🛆 of the European Union

0 () 000

П

COLOUR SENSOR

ABSTRACT

This activity is a playground to explore both colour recognition techniques and servo motors controls. Both the direct use of a newly released colour sensor and light reflection measurements are discussed. In the former case the traffic light is realized with a led shield, while in the latter one mBot, equipped with servo control, stands still and acts as the traffic lights by rotating a coloured disk. The main robot, instead, reads the traffic light colour and move forward or stop accordingly.

DIDACTIC OBJECTIVES

While playing the activity you will learn about:

- Physics: Light spectrum and Light measurements in reflective geometry.
- Technology: the servo motor control and the colour sensor.
- Computer Science: conditioned actions.
- Engineering: Automatic driving, simulating the recognition of a traffic light.

While implementing the code you will learn about:

- Computer Science: algorithm development.
- Mathematics: rotation angles

 STEM Subject:
 Science ⊠
 Technology ⊠
 Engineering ⊠
 Mathematics ⊠

 Education Level:
 12-14 years ⊠
 14-16 years ⊠

PROBLEM STATEMENT

Some students do not know the composition of the colors, which with the primary and secondary colors.

A robot runs along any given path, reads the traffic light colour and move forward or stop accordingly. In one setup (configuration a) the traffic light is a Arduino led shield and the colours are detected through a colour sensor. In configuration b, instead, one mBot, equipped with servo control, stands still and acts as the traffic light (by rotating a coloured disk) and the colour detection is performed through optical measurements in reflective geometry.

BOM (Bill of Materials Needed)

mCore of mBot



mBot Ranger



Colour Sensor



➢ Line Follower





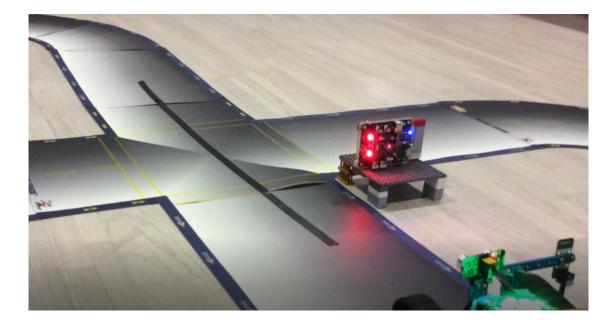
				P	PORT 1		Р	PORT 2		PORT 3			PORT 4		P.MOT	P.MOT2		
ELEMENT	ID	CABLE	E AMOUNT	Y	в	w	Y	В	w	γ	в	w	BI	Y	в	wв		W*
Mbot Robot 2'4G			1															
Motor 1	W*																W*	
Motor 2	W*																	W *
Me RJ 25 adapter	Υ																	
	В																	
	Bl																	
Mini Pan-Tilt kit																		
It has 2 servos.																		
We have to connect the servo to a RJ25 adapter																		
Mini Gripper																		
We have to connect the servo to a RJ25 adapter																		
Me 7-Segment serial display	В																	
Me Led Matrix 8x16	В																	
Me Ultrasonic sensor	Y																	
Me Temperature Sensor - Waterproof	Y																	
Me Line Follower	В	(1)	1		В													
Me Flame sensor	BI																	
Me PIR Motion sensor	В																	
Me Sound sensor	Bl																	
Me Touch sensor	В																	
Mini Fan Pack	В																	
Me Color Sensor	В	(1)	1					В										
Me Temperature and Humidity sensor	Υ																	
Me 130 Motor Fan Pack	В																	
RJ25 cables			2															
Structures and beams																		
Laptops																		
Attrezzo (not essential)																		



ACTIVITY DESCRIPTION

First version

The activity consists of drawing a circuit with black tape, create a traffic light with a mCore shield and thanks to the colour sensor, when the robot detects the red light it will stop, with the yellow light it will considerably decrease its speed and when it detects the green light will continue the speed or will be launched.



The colour is the property possessed by an object of producing different sensations on the eye as a result of the way it reflects or emits light.

The White light when decomposing organites the 7 colours of the visible spectrum: **red**, **orange**, **yellow**, **green**, **blue**, **cyan** and **violet**.

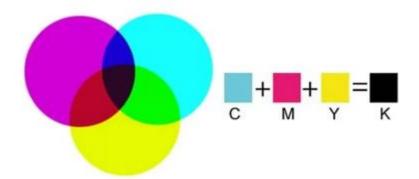
The colour and light are related and there are 2 theories that explain that relationship:

- 1. Substractive Synthesis:
 - > We treat colour as pigment and the primary colours are:





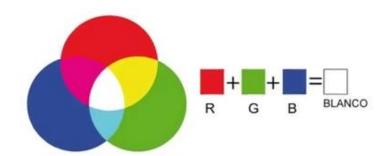
> If these colours are mixed the black colour is obtained:



- 2. Additive Synthesis:
 - > The primaries in light colour are:



> With these three colours, white is obtanied:



Next, the new colour sensor is detailed:

- It is a sensor that has recently come to the market.
- It is capable of recognizing up to 6 colours: Black, White, Red, Blue, Green and Yellow.
- > We need to install the library on mBlock:

MeColorSensor_Ranger 🔻 ———

(Arduino Mode ONLY)color sensor

(Arduino Mode ONLY)color sensor

MeColorSensor_mBot 🔻 —

(Arduino Mode ONLY)color sensor

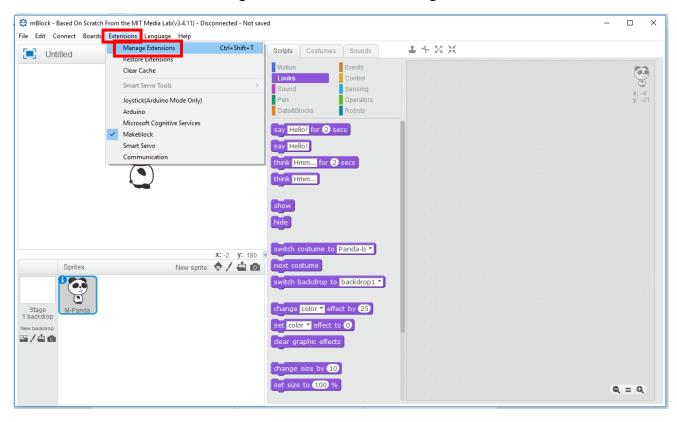
(Arduino Mode ONLY)color sensor



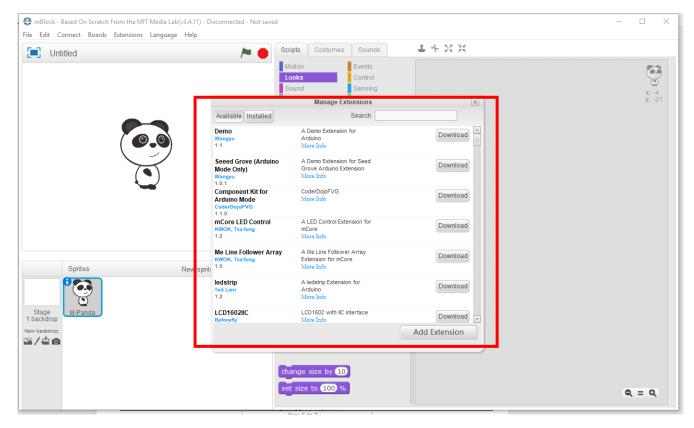
> Nowadays, it only works if we upload the code on the board.

For install the colour sensor library, we follow the next steps:

1. Start the mBlock software and go to "Extensions" => "Manage Extensions":



2. A new window will appear:



корон б 7 3. Insert the "color" word in the searcher:

8

	Manage Extensions	X
Available Installed	Search color	
Color sensor Flek 1.0	Extension for color sensor based on TCS34725 More Info	Download
TC S230 Color Sensor Gazihan Alankuş 1.0	TCS230 Color Sensor. Both Scratch and Arduino mode. More Info	Download
MeColorSensor_mBot Makeblock HK 1.6	MeColorSensor Extension for mBot More Info	Download
MeColorSensor_Ranger Makeblock HK 1.1	MeColorSensor Extension for Ranger More Info	Download
Me Color Sensor Makeblock 1.0	a color sensor for Makeblock More Info	Download
TFT_LCD (Arduino Mode Only) EnjoyneerHK 1.0.1	TFT_LCD: (0,0) at left bottom corner & 64 colors More Info	Download
		Add Extension

- 4. The "MeColorSensor_mBot" and the "MeColorSensor_Ranger" will appear. Now, click on "download".
- 5. If everything went well, the installed libraries will appear in the "robots" section.

File Edit Connect Boards Extensions Language	e Help			
Untitled	/ Scripts	Costumes Sounds		
	Motion	Events		6
	Looks	Control		
	Sound	Sensing		x: -4 y: -21
	Data&Bl	Operators ocks Robots		
	Me Color S			
		nsor Port1 R value		
	Color se	nsor Port1 detected	white	
	MeColorSe	nsor_mBot ▼	- • Internet and a second and a second	
	(Arduino	Mode ONLY)color sense	pr (Port	
	(Arduing	Mode ONLY)color sens	or Po	
	MeColorSe	nsor_Ranger▼	- •	
	(Arduino	Mode ONLY)color sense	pr Porté	
	x: 0 y: -180 🔍 (Arduing	Mode ONLY)color sens	or Port	
Sprites	New sprite: 🔶 🖊 🚢 🧰 Color sens	or 🗸		
963				
		olor sensor (Port6) , no		
Stage M-Panda		component Port6 , R		
1 backdrop		component Port6 , H		
	color or	Port6 is RED		
	mBot 🔻 —		- • Internet and a second and a second	
	mBot Pro	ogram		
	rup forw	ard 🔻 at speed 💽		Q = Q

Now, we develop the traffic light code for the mCore shield:

G	nBotProgram
	orever and a state of a
	set led on board all red 150 green 0 blue 0
	wait 10 secs in the hadrand a hadrand a hadrand
	set led on board all' red 0' green 150 blue 0'
	wait 10 secs
	set led on board all' red 255 green 255 blue 0
	wait 2 secs

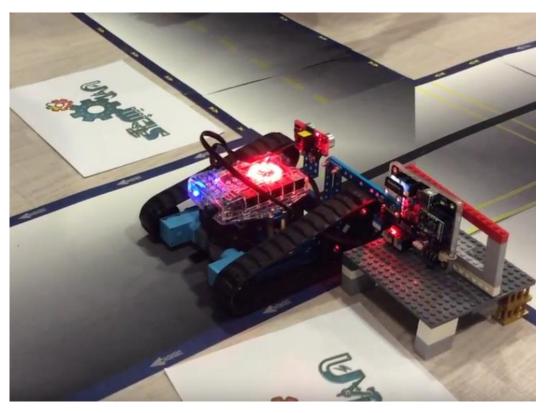
To finish, we will program the mBot Ranger to detect the colour.

When it detects the red colour, the Ranger will stop; when it detects the yellow light its speed will decrease and when the green light is active it will advance.

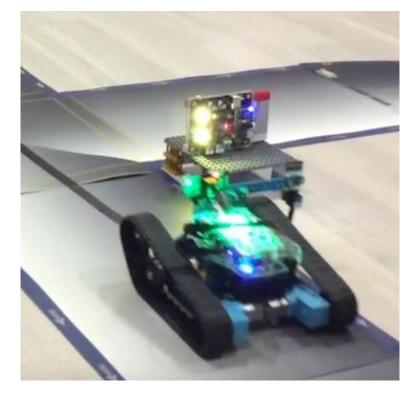
set led on board all? red () green () blue ()
forever
if (ArduinoMode ONLY)colorsensor Port8 detected red then
red
if (Arduino Mode ONLY)color sensor Ports detected yellow then
yellow and a standard
if (ArduinoMode ONLY)colorsensor Ports detected green then
green
define red
set red to (ArduinoMode ONLY)colorsensor Port8 R value
set led on board all? red red green green blue yellow
run forward T at speed 0
define green
set green 🔨 to (Arduino Mode ONLY)colorsensor Port® 🚱 value
set led on board all? red red green green blue yellow
run forward at speed 150
define yellow
set yellow to (ArduinoMode ONLY)colorsensor Port® B value
set led on board all? red red green green blue yellow



When Ranger detects the red colour it stops.



And if the colour is yellow its speed is lower and if the colour is green the mBot Ranger will be continue.





Second version

In the following, we list the simple steps needed to play the activity, discuss the colour detection technique and illustrate the algorithm and code.

Through the text, useful tips leading to better results are mentioned and highlighted in orange colour.

Experimental Procedure

- 1. Set up the "road": a track of black tape the mBot can follow.
- 2. Set up the Traffic Lights:
 - a. Install the Arduino Led shield on a proper support.
 - b. Choose one mBot to act as the traffic light. Mount on it the RJ25 adapter and the micro servo control. In our case, we installed the servo facing downward. Prepare a paper disk approximately 15 cm in diameter, divide it into three coloured sectors: yellow, green, red (or just two, discarding yellow). Make a hole in the disk center and mount it securely on the servo control.
- 3. Upload the code to the mBot(s)/mBot Ranger acting as the car(s) and to the traffic lights (mbot or shield).
- 4. Play the game! When the mBot reach the traffic light and if it is...

GREEN it will continue to run forward.

YELLOW it will slow down its speed, but continue to run forward.

RED it will stops at the traffic lights and it will continue to detect colours, in order to start again when the traffic lights turn green.

Light and Colour detection in reflective geometry (configuration b)

In physics, we speak of *a scattering experiment* when a given radiation (eg. light, as in our case) is sent to the investigated sample/surface and the scattered radiation intensity is measured. This can be done either in: *Transmission geometry*, where one measures the radiation that has gone through the sample and has emerged on the other side¹; or *Reflective geometry*, where one detects the radiation that is reflected by the sample and comes back almost in the same direction of the incident beam. This configuration is also known as *backscattering*.

If we recall the mechanism of our colour vision, it is easy to understand that Colours can be detected by light backscattering: an object appears for example green, because it reflects green light and absorb blue and red light.

¹ with respect to the incident beam



Therefore if we hit the object with red light, we will have a low reflected intensity, while if we use green incident radiation we will get an high scattered intensity. More in general each colour will reflect mostly light of the same colour; in case of a white object all colours will be reflected with high intensity, while a black object would absorb most radiation.

To perform the colour measurement, we turn on alternatively red, blue and green light on the led module and with the light sensor module we detect the reflected intensity for each colour. The software then elaborate the results.

For the measurement to work properly, it is necessary to embrace the light sensor into a small black tube (we used standard heat shrink tubing for electrical wires) in order to reduce the noise coming from the environment (eg. sunlight).

mBlock scratch Code for Colour Recognition

The following code control the Led and Light Sensor modules to perform the optical reflection measurements.

forever set led Port4 all red 255 green 0 blue 0 wait 0.2 secs set R to light sensor Port3 wait 0.2 secs set led Port4 all red 0 green 255 blue 0 wait 0.2 secs set G to light sensor Port3 wait 0.2 secs set led Port4 all red 0 green 0 blue 255 wait 0.2 secs set B to light sensor Port3 CheckBWsum if Color = 1 then MatchColor if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150	w	hen 🏓 clicked
<pre>wait 0.2 secs set R to (light sensor Port3) wait 0.2 secs set led Port4 all red 0 green 255 blue 0 wait 0.2 secs set G to (light sensor Port3) wait 0.2 secs set led Port4 all red 0 green 0 blue 255 wait 0.2 secs set B to (light sensor Port3) CheckBWsum if</pre>	fo	rever
<pre>wait 0.2 secs set R to light sensor Port3 wait 0.2 secs set led Port4 all red 0 green 255 blue 0 wait 0.2 secs set G to light sensor Port3 wait 0.2 secs set led Port4 all red 0 green 0 blue 255 wait 0.2 secs set B to light sensor Port3 CheckBWsum if</pre>	ſ	set led Port4 all red 255 green 0 blue 0
<pre>wait 0.2 secs set led Port4 all red 0 green 255 blue 0 wait 0.2 secs set G to light sensor Port3 wait 0.2 secs set led Port4 all red 0 green 0 blue 255 wait 0.2 secs set B to light sensor Port3 CheckBWsum if Color = 1 then MatchColor if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then </pre>		
<pre>wait 0.2 secs set led Port4 all red 0 green 255 blue 0 wait 0.2 secs set G to light sensor Port3 wait 0.2 secs set led Port4 all red 0 green 0 blue 255 wait 0.2 secs set B to light sensor Port3 CheckBWsum if Color = 1 then MatchColor if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then </pre>		set R to light sensor Port3
<pre>set led Port4? all? red 0 green 255 blue 0 wait 0.2 secs set G to light sensor Port3? wait 0.2 secs set led Port4? all? red 0 green 0 blue 255 wait 0.2 secs set B to light sensor Port3? CheckBWsum if Color = 1 then MatchColor if Color = 0 then set led on board all? red 0 green 0 blue 0? if Color = 2 then set led on board all? red 150 green 150 blue 150</pre>		
<pre>wait 0.2 secs set G to light sensor Port3 wait 0.2 secs set led Port4 all red 0 green 0 blue 255 wait 0.2 secs set B to light sensor Port3 CheckBWsum if Color = 1 then MatchColor if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150</pre>		
<pre>set G to light sensor Port3 wait 0.2 secs set led Port4 all red 0 green 0 blue 255 wait 0.2 secs set B to light sensor Port3 CheckBWsum if Color = 1 then MatchColor if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150</pre>		
<pre>wait 0.2 secs set led Port4 all red 0 green 0 blue 255 wait 0.2 secs set B to light sensor Port3 CheckBWsum if Color = 1 then MatchColor if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150</pre>		
<pre>set led Port4 all red o green o blue 255 wait 0.2 secs set B to light sensor Port3 CheckBWsum if Color = 1 then MatchColor if Color = 0 then set led on board all red o green o blue o if Color = 2 then set led on board all red 150 green 150 blue 150</pre>		
<pre>wait 0.2 secs set B to light sensor Port3 CheckBWsum if Color = 1 then MatchColor if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150</pre>		
<pre>set B to light sensor Port3 CheckBWsum if Color = 1 then MatchColor if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150</pre>		set led Port4 all red 0 green 0 blue 255
CheckBWsum if Color = 1 then MatchColor if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150		wait 0.2 secs
<pre>if Color = 1 then MatchColor if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150</pre>		set B 🔻 to light sensor Port3
MatchColor if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150		CheckBWsum
MatchColor if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150	ľ	if Color = 1 then
if Color = 0 then set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150		
set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150		Macheolor
set led on board all red 0 green 0 blue 0 if Color = 2 then set led on board all red 150 green 150 blue 150		if Color = 0 then
if Color = 2 then set led on board all red 150 green 150 blue 150		
set led on board all red 150 green 150 blue 150		set led on board all red or green or blue or
set led on board all red 150 green 150 blue 150		if Color - D than
		set led on board all red (150 green (150 blue (150)
		wait 1 secs



The CheckBWsmu function check if the total intensity is below or above certain thresholds, in order to detect Black and White respectively. These thresholds were calibrated during the experiment and might need (slight) recalibration for one's own experimental setup.

if R + set Color ▼ else if R + set Color	6			1300		· 	
else if R+	6		B	< 6		•	
if R+	G	+ (B				
	G	+	B	10			
set Color				< 00	00	the	en
	▼ to	0					
else							
set Color	▼ to	1					

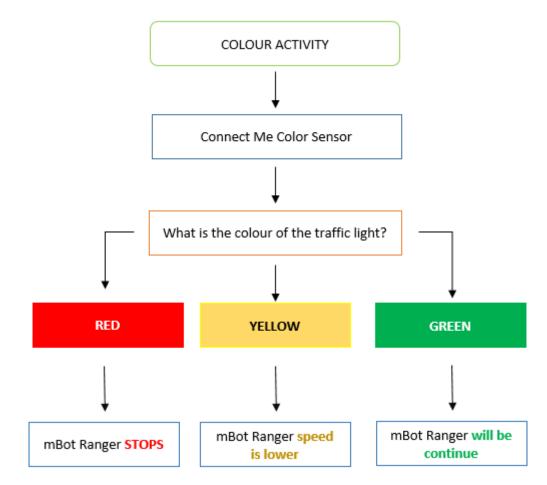
While the MatchColor function is the one which actually compare the measured intensities and output the final result by turning on the onboard led to the detected colour.

define MatchColor
if R > B and R > G then set led on board all red 255 green 0 blue 0
if $G > R$ and $G > B$ then
set led on board all ▼ red 0 ▼ green 255 ▼ blue 0 ▼
if B > R and B > G then set led on board all red Or green Or blue 255



FLOW CHART

Schema of the First Version activity:



STUDENT'S EVALUATION

Indicators for student evaluation may include:

- Physics: She/He performs laboratory measurements with care and accuracy.
- Physics: She/He is able to understand and compare experimental results.
- Physics: She/He properly understand the principles of light scattering.
- Physics: She/He recognizes.

BIBLIOGRAPHY

https://www.makeblock.es/productos/sensor color/

- [1] Me Led module description <u>http://learn.makeblock.com/me-rgb-led/</u>
- [2] Me Light sensor description <u>http://learn.makeblock.com/me-light-sensor/</u>



SCALABILITY

The activity is suitable for students aged 10 or higher.

15

The principles of optical spectroscopy (configuration b) and increasing details on the code algorithm might be explained only to older students (15-16 years).