Comunidad de Madrid
CONSEJERÍA DE EDUCACIÓN, JUVENTUD Y DEPORTE
Dirección General de Innovación, Becas y Ayudas a la Educación

PLAN DE FORMACIÓN EN LENGUAS EXTRANJERAS 2017

IN-57 Technology for ESO: Contents and Strategies

Lesson Plan: Traffic light with Arduino using code, S4A and Ardublock

Course 3rd ESO – Technology, Programming and Robotic

David Lobo Martínez

1. TOPIC

Arduino is an open source hardware platform, based on a simple board with analog and digital inputs and outputs, in a development environment that is based on programming languages. It is a device that connects the physical world with the virtual world, or the analog world with the digital world.

It will be a very easy project, but also practical and visual that will serve as an introduction to the Arduino board and its programming, in particular will be the control of a traffic light with LEDs.

The LEDs will light up successively for intervals of time, first the green light will go on for 15 seconds, then the yellow for 3 seconds and finally the red for 15 seconds, then this sequence will be repeated indefinitely.

The development of the program will be done with different types of software related to Arduino.

The execution will be performed physically on a breadboard connected to an Arduino card and also simulated.

2. AIMS:

To built a Traffic light with Arduino using different systems of programming:

- Arduino IDE.
- Simulation.
- S4A
- Ardublock

Also:

- Know how to use the Arduino card.
- Review basic connections.
- Know the interfaces of any of the controllers used in the technology workshop.
- Learn the basics of language for Arduino.
- Present the block diagram of a computer control system.

- Review the digital signal concept.
- Display the basic actions that can be performed with a computer control: driving LEDs.
- Show results physically and simulated by the computer.
- Know different types of software related to Arduino.

3. AGE GROUP:

   13 y.o

4. LEVEL:

   3rd grade of ESO

5. TIME:

   6 hours

6. MATERIALS:

- Arduino Uno and cable for USB connection.
- Personal computer with: Arduino software, S4A and Ardublock, and internet (online simulator www.tinkercad.com).
- 3 220 ohm resistors.
- 1 LED, green color.
- 1 LED, yellow color.
- 1 LED, red color.
- 1Breadboard.
- Connection cables.

7. PROCEDURE:

7.1 LEAD-IN/MODELING (1st hour)

Warm up with a video clip for introduction and some questions.
Show two Arduino project videos:
Ask students to share what they have observed
Introduction to Arduino hardware:
Learn about the main features of the board; microcontroller chip, USB connector, power jack slot, digital and analog input pins, output pins and reset pin
Discuss IDE (Integrated Development Environment) and Sketch

- Input Area - white area
- Status Bar - blue area below the white one
- Program Notification Area - black area below the blue one

Describe the functions of the 5 icons above the Input area
Ask students to connect the board to the laptops:
Ensure the Port connected to the Arduino is recognised
[Tools bar // select 'Port' // select Arduino Uno (Each Arduino board has a unique reference)]

7.2 TASK 1 (2nd hour)

Explaining Autodesk circuits, Sign up and first simulation (personal activity or in pairs).
Question quiz about introduction to Autodesk Circuits
Describe breadboard, LED, resistor, jumper wires
https://youtu.be/OCmeyJA5IRk?list=PLu8TYSQ5jCFho31LxXCoEBlL3x94l6mLc

7.3 TASK 2 (3rd hour)

Simulating traffic light in Autodesk Circuits (personal activity or in pairs).
Code is given by the teacher.

CODE:

Arduino uses its output pins to send electrical signals. These signals can be used to power other devices (led's, buzzers ...) or to communicate by sending a series of pulses that the receiver must understand.

The digital outputs are divided into two levels or states: HIGH to 5 V and LOW to 0 V.

By default, and without any program loaded on our controller card, the outputs are LOW level.

For constant values of 5V, 3.3V and 0V (GND) we can use the corresponding pins in the "POWER" section of our Arduino.

The pins that we can configure to change the level of output according to our program are the digital ones, that is, the 1 to 13.

By default these pins are configured as inputs so the first thing we have to do to use a pin as output is to configure it as such. For this we write this sentence in the section setup of the program.

PinMode (pin, OUTPUT);

Where we will replace "pin" with the number of the pin that we want to configure as output.

Now we have it configured as output but it still follows 0 V, ie at LOW level. When at a certain point in the program we want that pin to be set to HIGH level we will write this sentence

DigitalWrite (pinout, HIGH);

Now we will have 5 V permanently in that pin unless in the program we insert

DigitalWrite (pinout, LOW);

In which case it will return to the value of 0 V. We can set the pins to HIGH or LOW as many times as we want.

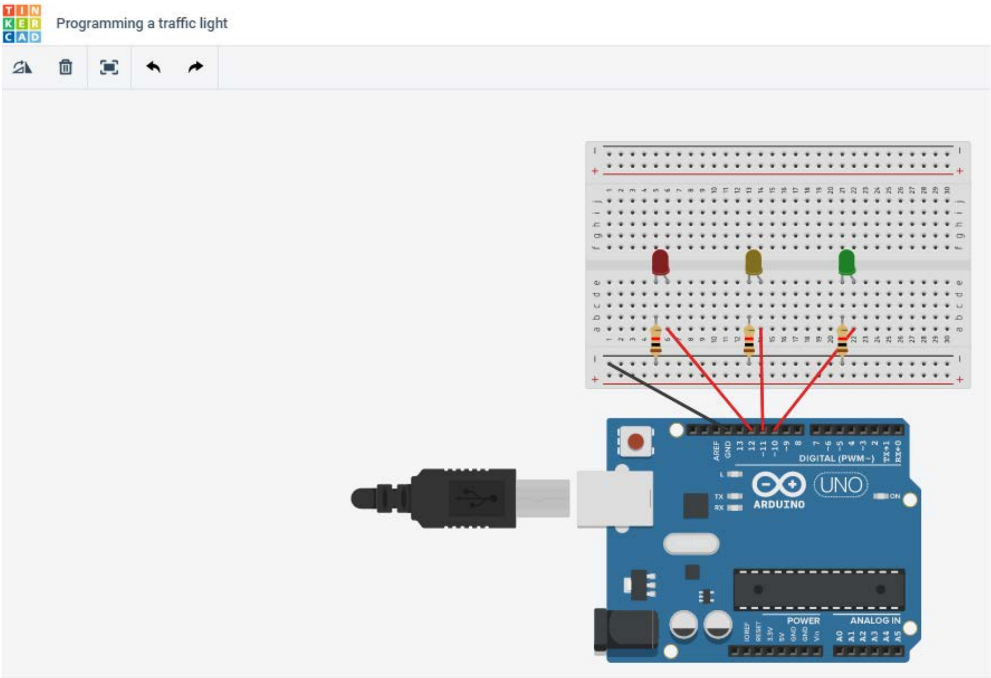
With the int statement we are declaring an integer variable, so that it can be used later in our code.

The delay command causes the program to stop for a certain amount of time. This will be defined, in milliseconds, within parentheses.

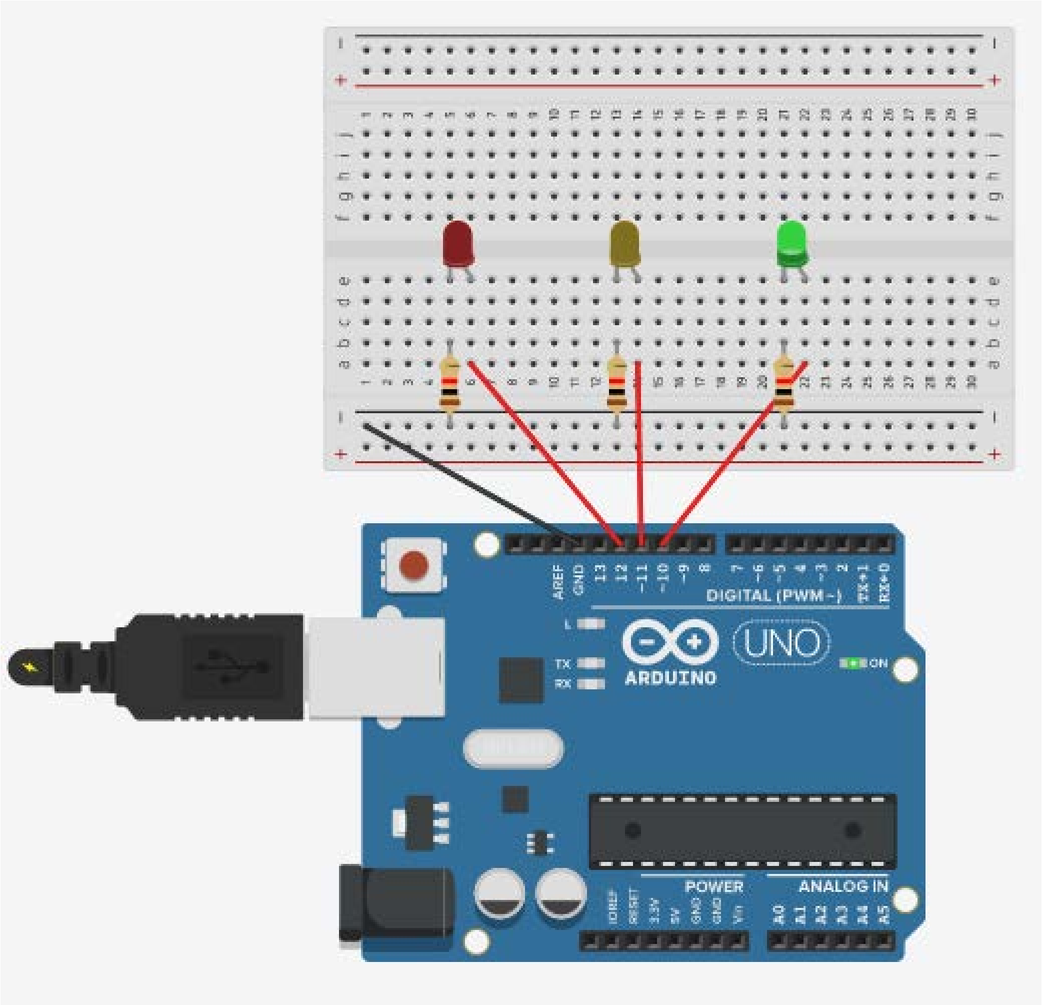
With these commands we can already program our code:

```
/******************/
/* TRAFFIC LIGHT */
/******************/
/*** David Lobo Martínez ***/

//** Definitions **//
int rojo=12; //It defines the value of the pin for the red led
int amarillo=11; //It defines the value of the pin for the yellow led
int verde=10; //It defines the value of the pin for the green led

//** Program **//

void setup() { //declarations
pinMode(verde,OUTPUT); //It declares the green pin as output
pinMode(amarillo,OUTPUT);//It declares the yellow pin as output
pinMode(rojo,OUTPUT); //It declares the red pin as output
}

void loop() { //repeat loop continuously
digitalWrite(verde,HIGH); //It turns on the green led
delay(15000); //wait 15 seconds
digitalWrite(verde,LOW); //It turns off the green led
delay(250); //wait 0.25 seconds

digitalWrite(amarillo,HIGH); //It turns on the yellow led
delay(3000); //wait 3 seconds
digitalWrite(amarillo,LOW); //It turns off the yellow led
delay(250); //wait 0.25 seconds

digitalWrite(rojo,HIGH); //It turns the red led
delay(15000); //wait 15 seconds
digitalWrite(rojo,LOW); //It turns off the red led
delay(250); //wait 0.25 seconds
}
```

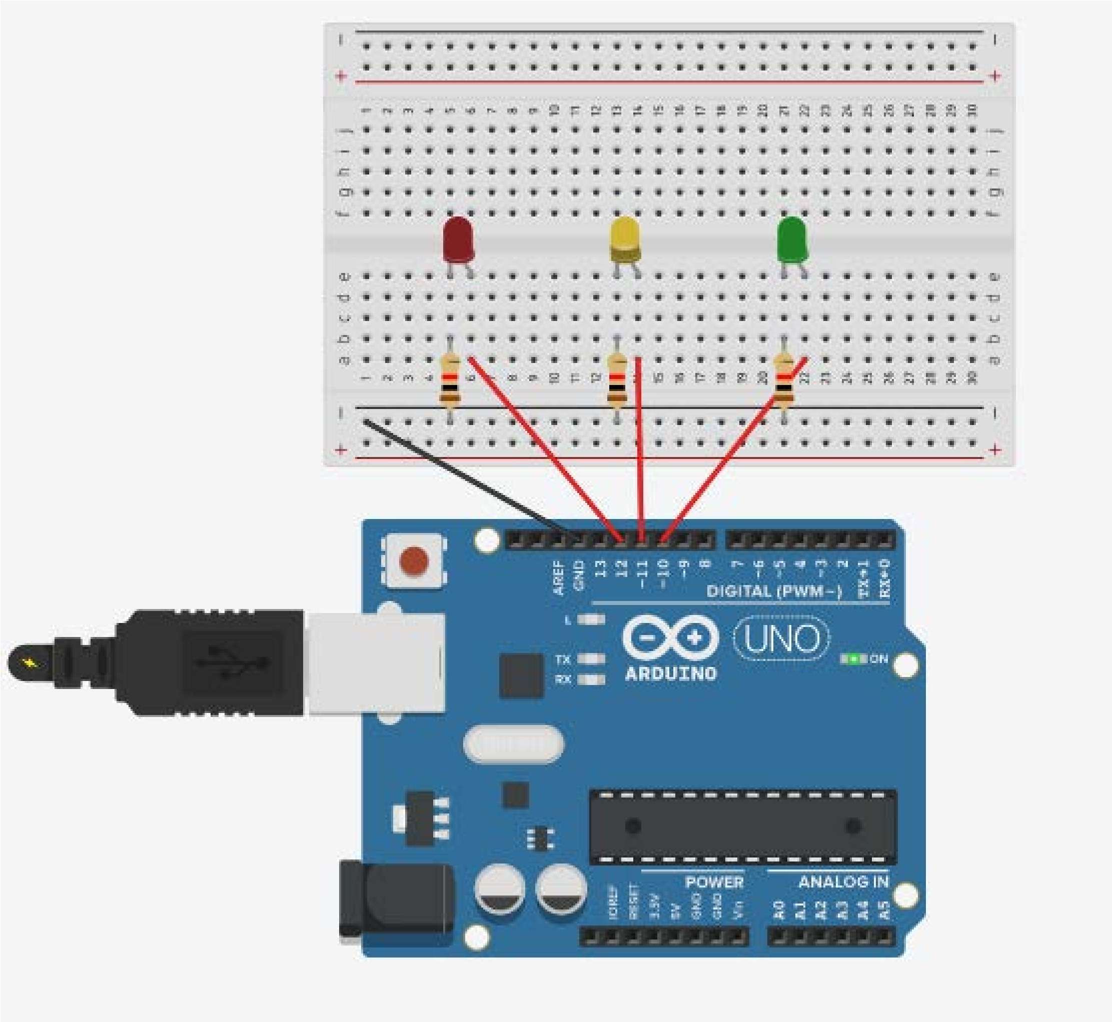
The set up will be the following:



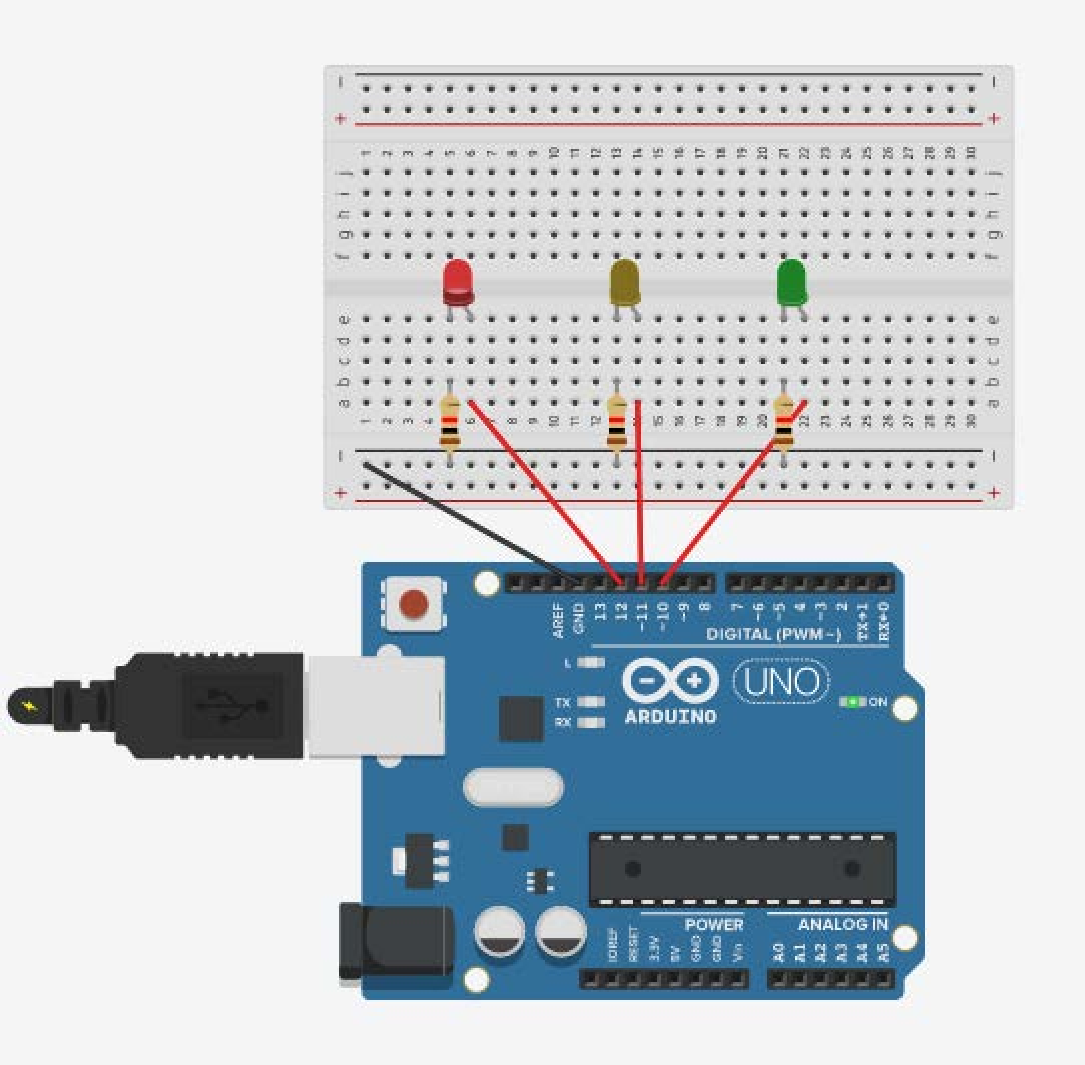
Green light turns on:



Yellow light turns on after 15 seconds:



Finally, red light turns on after 3 seconds:



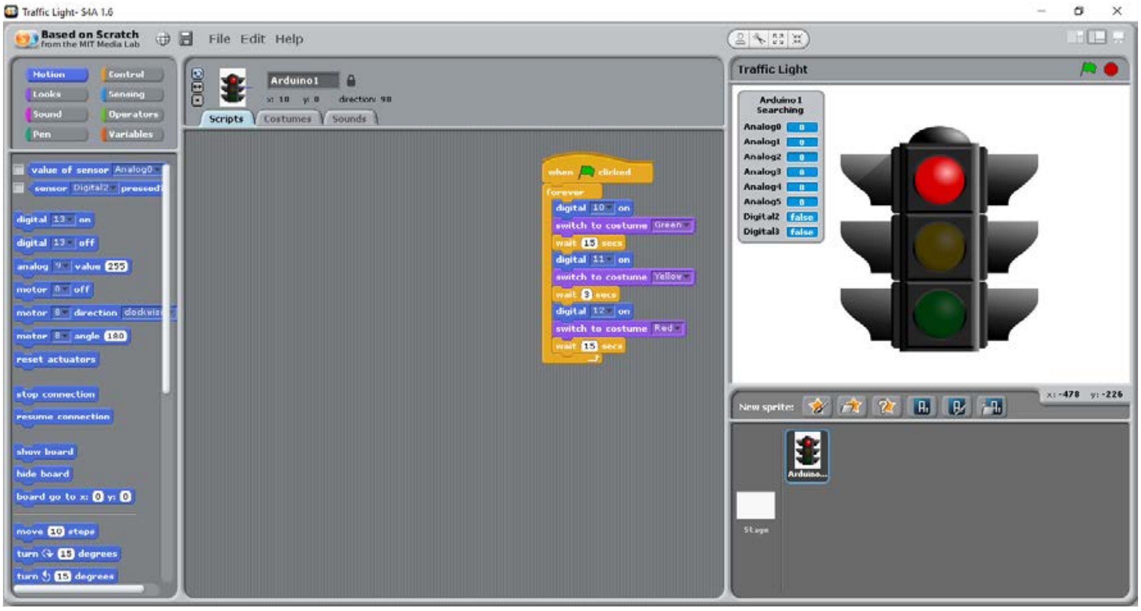
In this link you can see the simulation:
https://www.tinkercad.com/things/ieCzgIFnyGo-programming-a-traffic-light/

7.4 TASK 3 (4th hour)

S4A and Ardublock traffic light (personal activity or in pairs). Code is developed by students.

To use S4A, we must first install the S4A firmware on our Arduino board. Finally we will look for images of traffic lights in different colors to be able to use as scratch costumes:

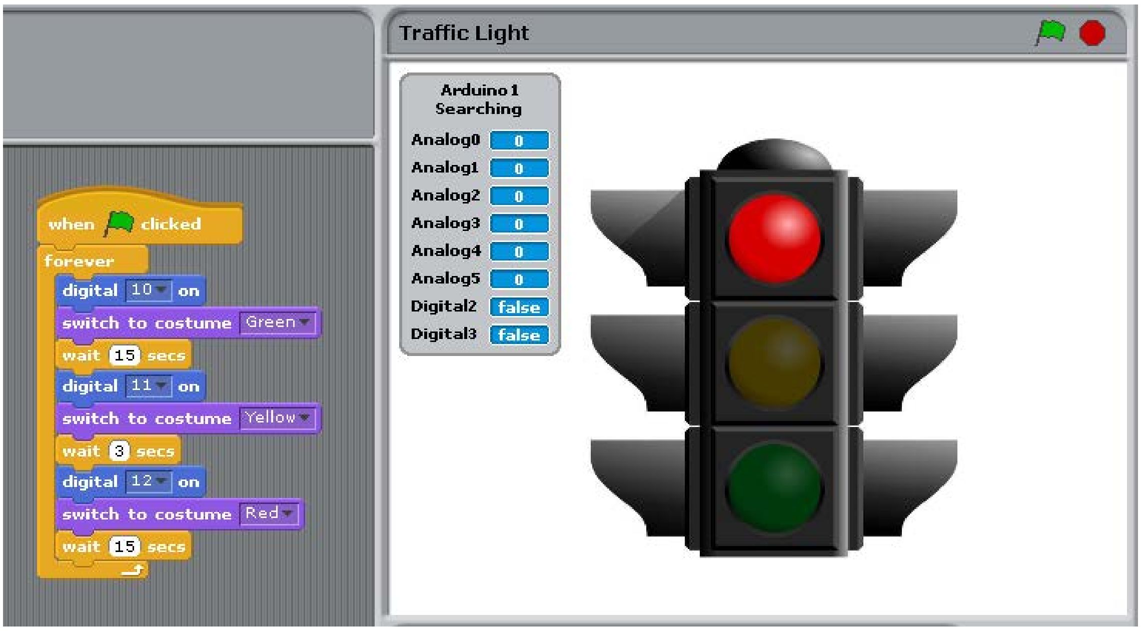
S4A screenshot:



```
when clicked
forever
    digital 10 on
    switch to costume Green
    wait 15 secs
    digital 11 on
    switch to costume Yellow
    wait 3 secs
    digital 12 on
    switch to costume Red
    wait 15 secs
```

Arduino1 Searching
Analog0 0
Analog1 0
Analog2 0
Analog3 0
Analog4 0
Analog5 0
Digital2 false
Digital3 false

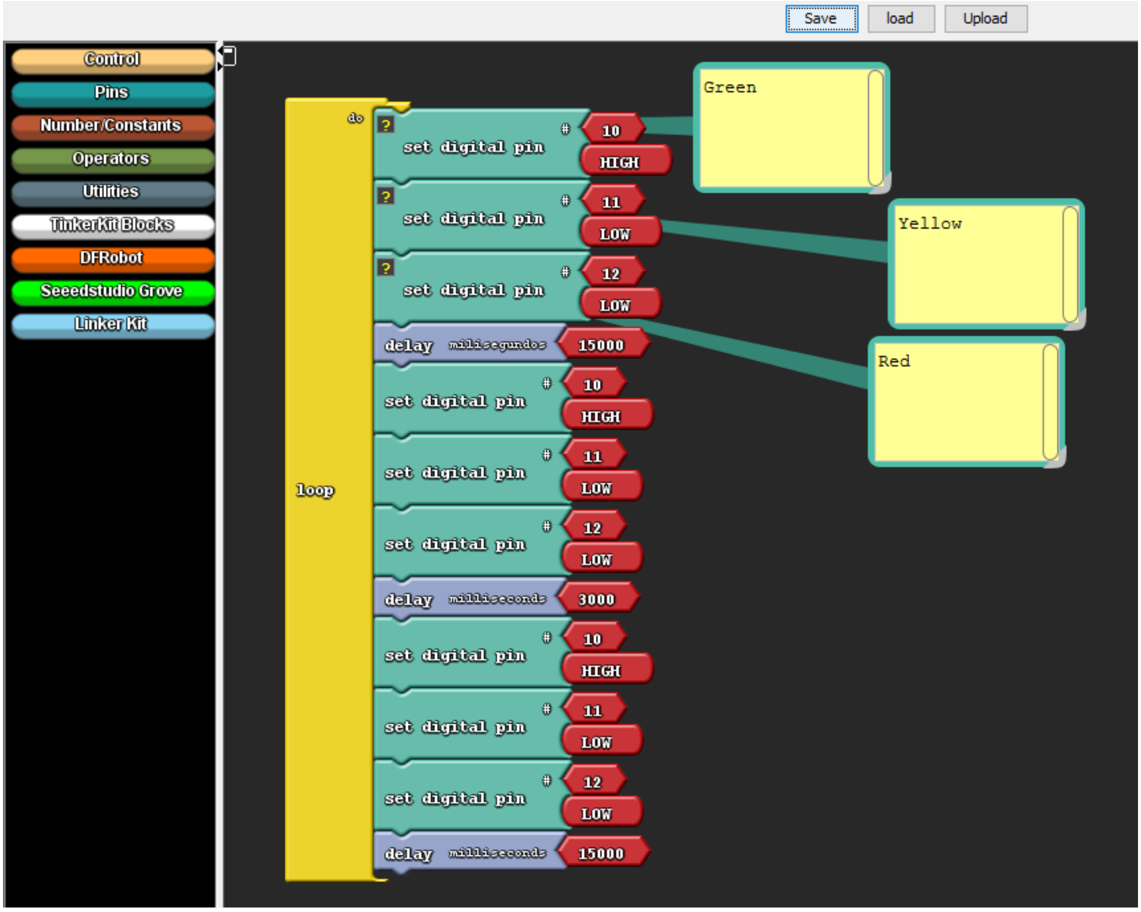
Blocks:



```
when ⚑ clicked
forever
    digital 10 on
    switch to costume Green
    wait 15 secs
    digital 11 on
    switch to costume Yellow
    wait 3 secs
    digital 12 on
    switch to costume Red
    wait 15 secs
```

```
Arduino1
Searching
Analog0   0
Analog1   0
Analog2   0
Analog3   0
Analog4   0
Analog5   0
Digital2  false
Digital3  false
```

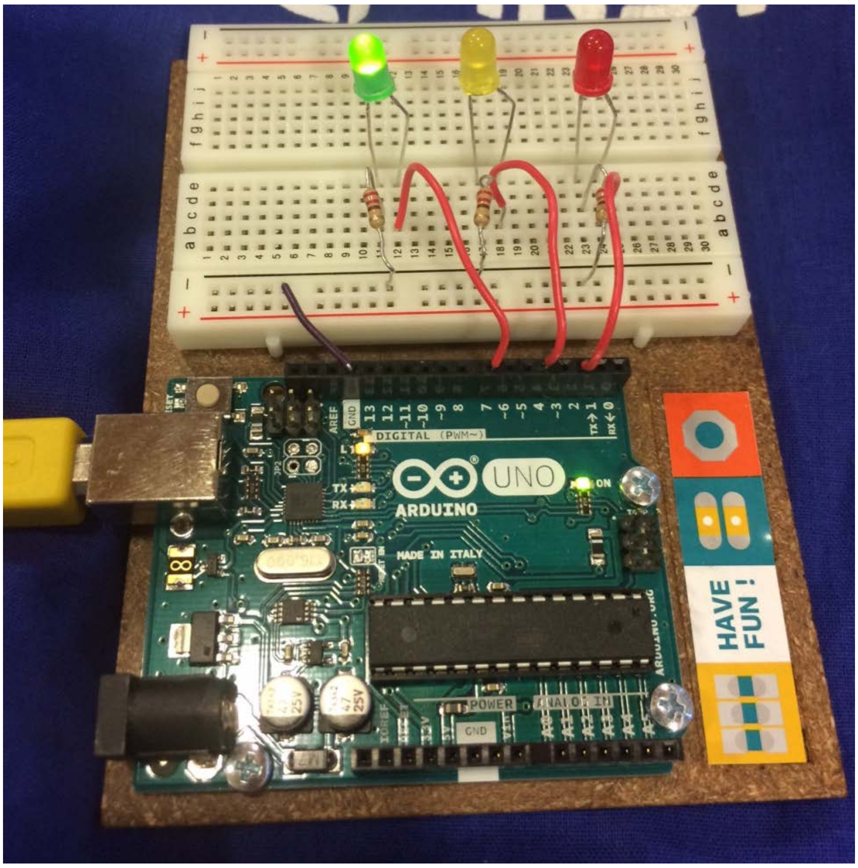
Ardublock code blocks to built a Arduino traffic light:



```
loop
do
    set digital pin   # 10   HIGH
    set digital pin   # 11   LOW
    set digital pin   # 12   LOW
    delay milisegundos 15000
    set digital pin   # 10   HIGH
    set digital pin   # 11   LOW
    set digital pin   # 12   LOW
    delay milliseconds 3000
    set digital pin   # 10   HIGH
    set digital pin   # 11   LOW
    set digital pin   # 12   LOW
    delay milliseconds 15000
```

7.5 TASK 4 (5th hour)

Set up Arduino traffic light in a real breadboard (group of 5) by code and connecting Scratch for Arduino.



7.6 TASK 5 (6th hour)

Final individual quiz and debrief in a paper sheet about the contents and objectives of the project .

OBJECTIVES

- Know the operation and use the Arduino controller card.
- Learn how to use flowcharts when performing programming tasks.
- Introduce the concept of Arduino.
- Show basic connections.

- Know the interfaces of any of the controllers used in the technology workshop.
- Learn the basics of language for Arduino.
- Present the block diagram of a computer control system.
- Review the digital signal concept.
- Display the basic actions that can be performed with a computer control: driving LEDs.
- Present a simple computer control system.
- Show results physically and simulated by the computer.
- Know different types of software related to Arduino.

CONTENTS

- Types of resistances.
- Calculation of the required resistance in a circuit.
- Semiconductors and diodes. Diodes LED.
- Mount simple electronic circuits.
- Design electrical and electronic circuits with appropriate software.
- Arduino, assembly and programming.
- Arduino simulator online.
- Programming with S4A.
- Programming with Ardublock.

8. RUBRIC

Items	Exceeds - 20	Fair-10	Poor- 0	Marks
Design – 20	The projects are exceptionally atractive	The project are acceptably atractive	The designs can be clearly improved	
It works- 30	All traffic light (simulation, S4A, ardublock and Breadboard) work	Some of the traffic light doesn't work	The traffic light doesn't work at all	
Question sheet - 20	Most of questions are right answered	Some questions are right answered	Most of questions are not right answered	
Individual Work and attitude.- 10.	The student works hard with a good attitude	The student fulfills the expectations	The student does not work well and even disturbs his colleagues.	

EVALUATION CRITERIA

- Distinguish the main input and output elements of a control system.
- Describe the characteristics of a controller, paying special attention to its outputs and inputs, both analog and digital.
- Use the controller to examine the operation of a system through the computer.
- Develop simple language control procedures for Arduino.
- Draw flow diagrams.
- Develop programs that control the inputs and outputs of a controller.
- Handle simple electronic circuits from a computer and a controller.