Traffic Lights with Arduino

1. INTRODUCTION

In this Project we are going to simulate the operation of a traffic light for vehicles in a automated by the action of a user or pedestrian through the action of a button. The main reason for the realization of this Project lies in the deep interest in the Arduino capabilities as an electronic control tool. The handling of Arduino implies a great reduction of time in electronic design and in the programming your microcontroller. Its characteristics make it a business opportunity in terms of its implementation in the sectors of automation and industrial communications, in order to a future dedication labor within the framework of these industrial sectors.

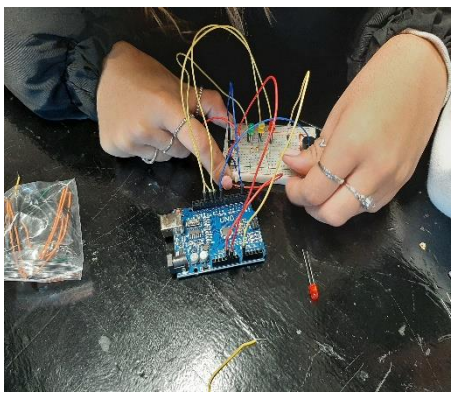
.

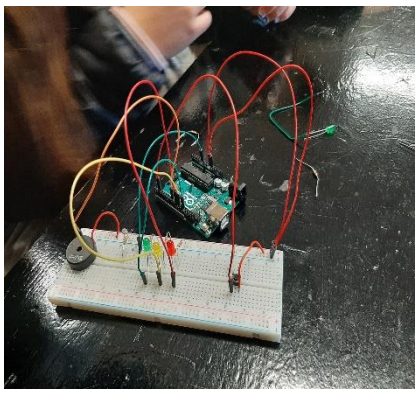
2. OBJETIVOS:

- The main objective of this project **is** familiarization with Arduino.

- Make connections of physical components on the board

- Program digital automatons: Arduino implementation in the control of groups of traffic lights that exemplifies a real traffic light system.



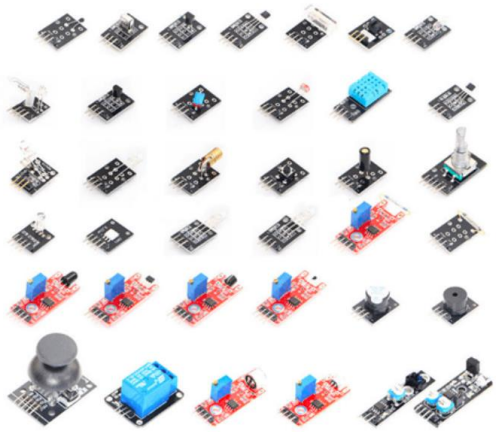


3. COMPONENTES

-Breadboard PCB
-LEDs colores
-1 Pulsador
-1 Timbre o buzzer
-1 Sensor Fotoluminoso
-Resistencias de 220Ω
-12 cables dupont

CALCULADORA RESISTENCIA:

https://www.digikey.es/es/resources/conversion-calculators/conversion-calculator-resistor-color-code

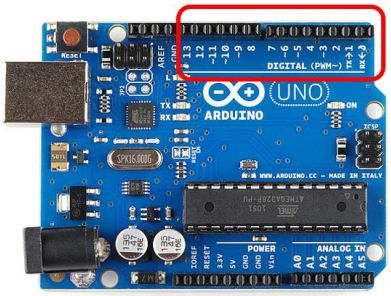


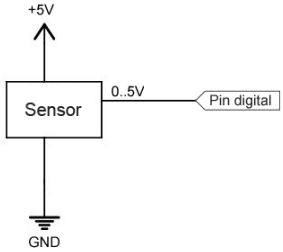
4. CONOCEMOS LA PLACA ARDUINO

In this project we are going to work with the Arduino uno board, specifically the R3. The main characteristics of this board is that they have on the one hand energy or power, on the other analog pins and finally 13 digital pins. Note that the energy will have values of 5v and 3.3v, in addition to its corresponding GND or Ground connection

Hardware:

DIGITAL PINS IN ARDUINO One of the most interesting functions (if not the most) of Arduino is to generate automata. This interaction takes place largely through the use of inputs and outputs. A digital signal is a voltage variation between -Vcc to +Vcc without going through intermediate values. Therefore, a digital signal has only two states, on and off.

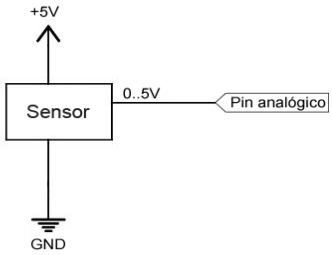




PIN ANALÓGICOS EN ARDUINO

Analog inputs work in a similar way to digital inputs, so in practice the assembly and final code are very similar. An analog signal is a magnitude that can take any value within a range –Vcc and + Vcc. For example, a voltage analog signal between 0V and 5V could be worth 2.72V, or any other value to any number of decimal places.

.



GND: DAR CORRIENTE DE FORMA CONTÍNUA

Software

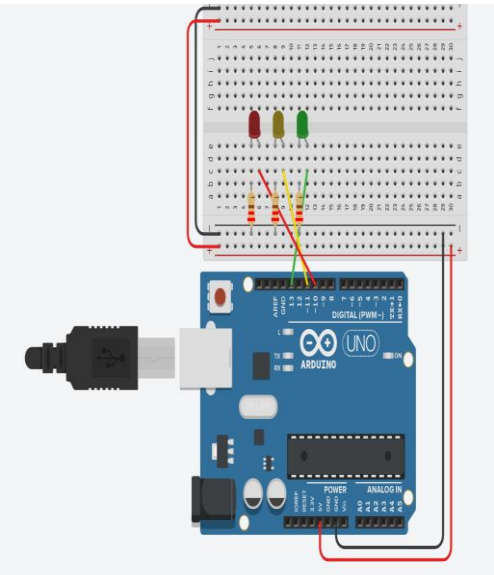
Now we will configure the two main functions of Arduino which are Setup and Loop.

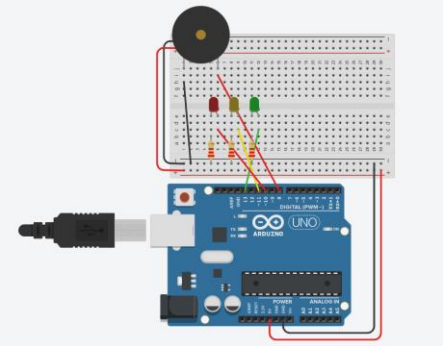
The Setup function will be in charge of initializing the parameters and pins with the default configuration. The Loop function This function is the one that is going to be repeated indefinitely. Here we will check the status of the buttons, we will turn on and off the multiple LEDs as appropriate

5. FASE MECÁNICA DEL PROYECTO

- Creation of Grafcet as a self-help to clarify the project assembly steps.
- Connection and programming of 3 LEDs.
- Buzzer connection and implementation.
- Mounting of a pushbutton as input pin.
- Futoluminous sensor capable of detecting light intensity.

We started our project with the mounting of 5 LEDs on our PCB board. 3 LEDs (Red, yellow and green) will be used to simulate the assembly of a traffic light for cars and on the other hand we will use 2 LEDs (Red and blue) to simulate.To do this we will connect the ends of the LED (anode and cathode) to each of the ends of the PCB. The anode or positive part of the diode will be connected to a digital pin. It must be connected to a digital pin since the value of this LED will at all times be activated or deactivated (that is, 0 or 1). The other end of the LED will be connected to ground or GND, although it must previously have a 220Ω resistor connected. In the same way, the connection of the other two LEDs will be made, each one to a different pin. Specifically in our project we will use pin 13, 11, 10 for the green, yellow and red LEDs. And we will use pins 7 and 5 (blue led and red led).





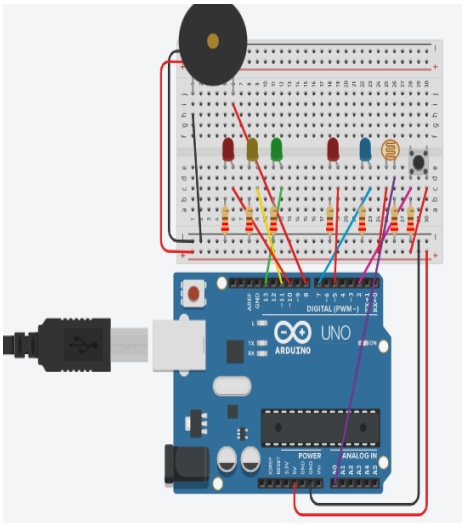
Next we mount the Buzzer on our board, the connection will be analogous to the diodes since the buzzer has two poles (positive and negative). The positive pole of the device will be connected to a digital pin of the arduino, specifically we will use pin 8. For the buzzer we will not use resistance to allow it to work at more frequencies by not losing electrical power.

.

Next we will mount the button on the PCB. We remember that the button has two states, (open or closed). Our objective will be to connect the button directly to 5V and GND to the other extreme as we can see in the following diagram, although we remember the use of a 220Ω resistor to regulate the voltage that passes through the circuit at all times. In the PCB pin, as if it were a multimeter, we will connect a cable to digital pin 2. This pin will be configured as an input since we need to capture when the button has been activated. We have used a digital pin since we only need to know if the button has been activated or not, but in no case control with what power or force it has been activated.

With this scheme we would have a traffic light capable of being automated with the action of a user or pedestrian. Given the current situation we are experiencing with COVID, an improvement to this project has seemed interesting to us. Specifically, to avoid the user having to physically press the button, we have added a photoluminous sensor or resistance to our model. As it is a diode, we connect it analogously to the LEDs with a 220 Ω resistor. This sensor will be connected to Analog pin 0, since in this case we need to know the specific values that the sensor is receiving.

Next we will show the final image or grafcet of the circuit



6. FASE LÓGICA DEL PROYECTO: PROGRAMACIÓN

As mentioned in previous sections, we are going to build two traffic lights (one for cars and one for pedestrians) which will be automatons. The traffic light of the cars will start green and the yellow and red LEDs will turn off. On the other hand, the pedestrian traffic light will start with the red led on and the blue one off.

- After activating the button or receiving the photoluminous resistance of a light-raising value (later we will specify the value) our project will carry out the following sequence:
- - The vehicle traffic light:
- o Turns off the green led and turns on the yellow for 1.25s.
- o Turn off the yellow led and turn on the red one for
- - The pedestrian traffic light :
- o Turns off the red LED and activates the Blue for
- - buzzer
- o During the time that the vehicle traffic light is red, the buzzer will emit an intermittent sound indicating to pedestrians that they can cross. The intermittent sound will increase the repetition frequency progressively, indicating that the ignition time is over.

- For LED programming, in the void setup() we will activate the digital pins through the function PinMode("pin number", OUTPUT). In our case, all the LEDs will be output elements and we will turn them all off except the green LED for the vehicle traffic light and the red one for the pedestrian traffic light. For this we will use the function digitalWrite ("PIN", HIGH). We will finish this function leaving the other pins activated, that is, the buzzer pin as input device, while the button and the light sensor will be input elements. pinMode("pin number", INPUT).
- In the void loop() function, we are going to perform the main function of the project. In our case we will read the sensor and the button. The button will be read with the function digitalRead("pin number") since it is a digital pin, on the other hand, for the light sensor we will call the function analogRead("analog pin number). The function will be composed of a conditional (IF / ELSE). In the IF we will use a compound conditional, in this case if the button is deactivated and the light sensor has a value less than the determined parameters), the traffic light is in its initial state. In another case (else) the steps described in the previous process will be activated sequentially.
- The light sensor will be conditioned to the lighting situation of the room or environment in which the project is located. To do this, we will previously add the serial.print() function in the loop, in this way in the serial port we will print the values that the sensor is receiving in the environment where we are going to use it. Once these parameters have been observed, if we verify, for example, that they are not greater than 100, it will suffice to add in the previous conditional that the determined parameter is not greater than 130 (a higher margin must be left to ensure that no unwanted light activates the traffic light).

CODIGO

```
int GREENCAR = 13;
int YELLOWCAR = 11;
int REDCAR = 10;
int BLUEPERSON = 7;
int REDPERSON = 5;
int BUZZER = 8;
int BUTTON = 2;
int LIGHTSENSOR= A0;
int PUSH;
int PASSLIGHT;
void setup() {
  // put your setup code here, to run once:
 pinMode (GREENCAR, OUTPUT);
 digitalWrite (GREENCAR, HIGH);
 pinMode (YELLOWCAR, OUTPUT);
 digitalWrite (YELLOWCAR, OUTPUT);
 pinMode (REDCAR, OUTPUT);
 digitalWrite (REDCAR, LOW);
 pinMode (BLUEPERSON,OUTPUT);
 digitalWrite (BLUEPERSON, LOW);
 pinMode (REDPERSON, OUTPUT);
 digitalWrite (REDPERSON, LOW);
 pinMode (BUZZER,OUTPUT);
 pinMode(BUTTON, INPUT);
 Serial.begin(9600);

}

void loop() {
  // put your main code here, to run repeatedly:
```

```
  PUSH = digitalRead(BUTTON);
  PASSLIGHT = analogRead (A0);
  Serial.print("el valor de luz es "); // Para imprimir
  Serial.print(PASSLIGHT); //
  Serial.print("\n"); //
  if (PUSH == LOW && PASSLIGHT<100){
  digitalWrite(GREENCAR, HIGH);
  digitalWrite(REDCAR, LOW);
  digitalWrite(YELLOWCAR, LOW);
  digitalWrite(REDPERSON, HIGH);
  digitalWrite(BLUEPERSON, LOW);
}else{
  digitalWrite(YELLOWCAR, HIGH);
  digitalWrite(GREENCAR, LOW);
  digitalWrite(REDCAR, LOW);
  digitalWrite(BLUEPERSON, LOW);
  digitalWrite(REDPERSON, HIGH);
delay(1250);
digitalWrite(YELLOWCAR,LOW);
digitalWrite(GREENCAR, LOW);
digitalWrite(REDCAR, HIGH);
digitalWrite(BLUEPERSON, HIGH);
digitalWrite(REDPERSON, LOW);
tone(BUZZER,234);
delay(1000);
noTone(BUZZER);
delay(1000);
tone(BUZZER,234);
delay(700);
noTone(BUZZER);
delay(700);
tone(BUZZER,234);
delay(500);
noTone(BUZZER);
```

```
delay(500);
tone(BUZZER,234);
delay(300);
noTone(BUZZER);
delay(300);
tone(BUZZER,234);
delay(100);
noTone(BUZZER);
delay(100);
tone(BUZZER,234);
delay(100);
noTone(BUZZER);
delay(80);
tone(BUZZER,234);
delay(50);
noTone(BUZZER);
delay(30);
tone(BUZZER,234);
delay(4000);
noTone(BUZZER);
digitalWrite(REDCAR,HIGH);
digitalWrite(YELLOWCAR,LOW);
digitalWrite(GREENCAR,LOW);
digitalWrite(BLUEPERSON,LOW);
digitalWrite(REDPERSON,LOW);
delay(3000);



  }

}
```