

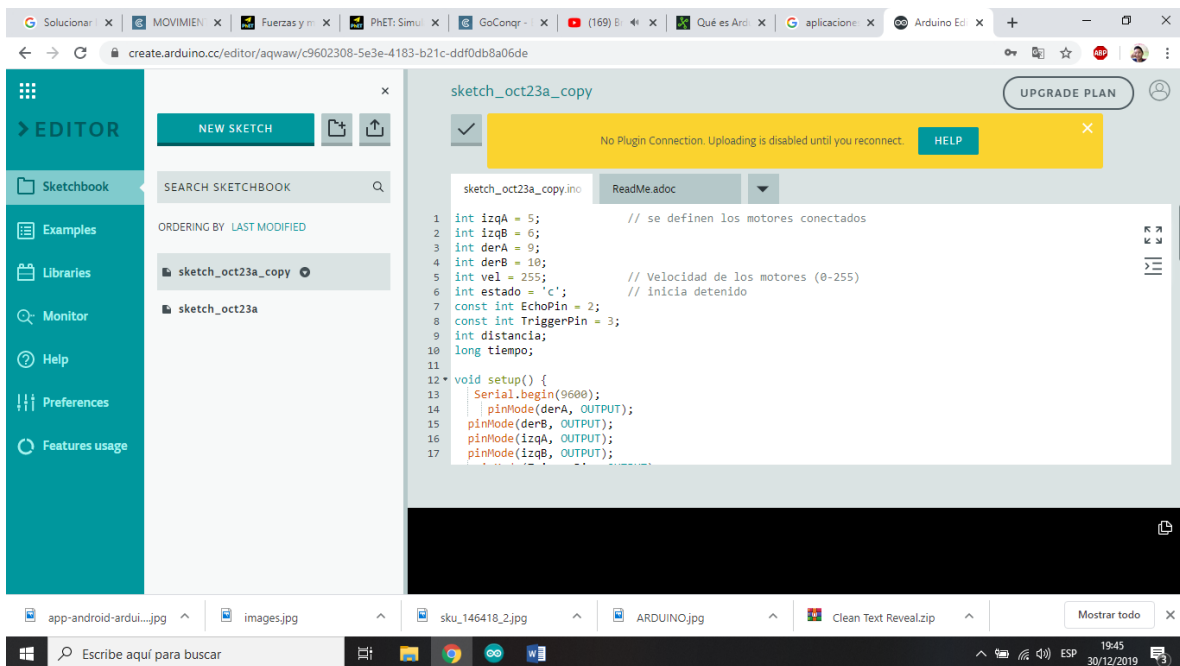
PROYECTO ARDUINO

Para realizar nuestro proyecto del movimiento rectilíneo uniforme en Arduino debemos introducir la siguiente programación en la aplicación Arduino Build o en su página oficial vía online:

<https://create.arduino.cc/editor/aqwaw/c9602308-5e3e-4183-b21c-ddf0db8a06de>

Esta programación nos permitirá mover el carrito previamente construido y de acuerdo a nuestra necesidad lo moveremos, ya sea para arriba, abajo, izquierda y abajo mediante bluetooth.

Dentro de la programación también está integrada la codificación para determinar la distancia recorrida por el móvil mediante el uso del sensor Ultrasonico, dato muy útil a la hora de obtener la velocidad del carrito.



```
sketch_oct23a_copy
1 int izqA = 5; // se definen los motores conectados
2 int izqB = 6;
3 int derA = 9;
4 int derB = 10;
5 int vel = 255; // Velocidad de los motores (0-255)
6 int estado = 'c'; // inicia detenido
7 const int EchoPin = 2;
8 const int TriggerPin = 3;
9 int distancia;
10 long tiempo;
11
12 void setup() {
13   Serial.begin(9600);
14   pinMode(derA, OUTPUT);
15   pinMode(derB, OUTPUT);
16   pinMode(izqA, OUTPUT);
17   pinMode(izqB, OUTPUT);
18 }
```

Aquí integraríamos la siguiente programación para que nuestro carrito pueda moverse por la aplicación Android.

```

int izqA = 5;          // se definen los motores conectados
int izqB = 6;
int derA = 9;
int derB = 10;
int vel = 255;       // Velocidad de los motores (0-255)
int estado = 'c';    // inicia detenido
const int EchoPin = 2;
const int TriggerPin = 3;
int distancia;
long tiempo;

void setup() {
  Serial.begin(9600);
  pinMode(derA, OUTPUT);
  pinMode(derB, OUTPUT);
  pinMode(izqA, OUTPUT);
  pinMode(izqB, OUTPUT);
  pinMode(TriggerPin, OUTPUT);
  pinMode(EchoPin, INPUT);
}

void loop() {
  if(Serial.available()>0){ // lee el bluetooth y almacena en estado
    estado = Serial.read();
  }
  if(estado=='a'){ // Boton desplazar al Frente
    analogWrite(derB, 0);
    analogWrite(izqB, 0);
  }
}

```

```
    analogWrite(derA, vel);
    analogWrite(izqA, vel);
}
if(estado=='b'){    // Boton IZQ
    analogWrite(derB, 0);
    analogWrite(izqB, 0);
    analogWrite(derA, 0);
    analogWrite(izqA, vel);
}
if(estado=='c'){    // Boton Parar
    analogWrite(derB, 0);
    analogWrite(izqB, 0);
    analogWrite(derA, 0);
    analogWrite(izqA, 0);
}
if(estado=='d'){    // Boton DER
    analogWrite(derB, 0);
    analogWrite(izqB, 0);
    analogWrite(izqA, 0);
    analogWrite(derA, vel);
}

if(estado=='e'){    // Boton Reversa
    analogWrite(derA, 0);
    analogWrite(izqA, 0);
    analogWrite(derB, vel);
    analogWrite(izqB, vel);
}
digitalWrite(TriggerPin, LOW); // se envia un pulso bajo al trigger
delayMicroseconds(5);
```

```
digitalWrite(TriiggerPin, HIGH); //se envía un pulso para activar el sensor
delayMicroseconds(10);
digitalWrite(TriiggerPin, LOW);

// medimos el pulso de respuesta
tiempo = (pulseIn(EchoPin, HIGH)/2);

// dividido por 2 por que es el
// tiempo que el sonido tarda
// en ir y en volver
// ahora calcularemos la distancia en cm
// sabiendo que el espacio es igual a la velocidad por el tiempo
// y que la velocidad del sonido es de 343m/s y que el tiempo lo
// tenemos en millonesimas de segundo

distancia = float(tiempo * 0.0343);
Serial.println(distancia);

delay(1000);

}
```