

**UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO**

**FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y  
MECÁNICA**

**ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS**

---

---

**CARRITO ESQUIVANDO PELOTITAS CON ARDUINO Y OPENCV**

---

---

**CURSO :**  
ROBÓTICA Y PROCESAMIENTO DE SEÑAL

**DOCENTE :**  
ING. JOSE MAURO PILLCO QUISPE

**ALUMNOS :**

BOLAÑOS CCOPA, RONY	154624
CALLASACA ACUÑA, FERNANDO	140989
LIMPE QUISPE, JERRY ANDERSON	140985
MENDOZA CJUIRO, NILO FIDEL	144987
SERRANO AMAU, FIORELA ELOISA	133968

**2020-I**

## Contenido

<b>Presentación.....</b>	<b>4</b>
<b>Introducción .....</b>	<b>5</b>
<b>1. Marco Teórico.....</b>	<b>6</b>
<b>1.1. Arduino Uno.....</b>	<b>6</b>
<b>1.2. 4WD Robot Chasis Kit.....</b>	<b>7</b>
<b>1.3. 4 motores DC Gear Motor 1:48.....</b>	<b>8</b>
<b>1.4. Neumático Rueda 65x26mm.....</b>	<b>9</b>
<b>1.5. Driver Puente H L298N .....</b>	<b>9</b>
<b>1.6. Batería 18650.....</b>	<b>12</b>
<b>1.7. Cables hembra y macho .....</b>	<b>13</b>
<b>1.8. Arduino 1.8.13.....</b>	<b>13</b>
<b>1.9. Cámara HUAWEI P30 lite .....</b>	<b>15</b>
<b>1.10. Bolitas de Plástico de diferentes colores .....</b>	<b>16</b>
<b>1.11. Python 3.8.6.....</b>	<b>17</b>
<b>1.12. Librerías .....</b>	<b>17</b>
<b>Numpy .....</b>	<b>17</b>
<b>Opencv.....</b>	<b>17</b>
<b>Serial.....</b>	<b>19</b>
<b>1.13. Módulos .....</b>	<b>20</b>
<b>Time.....</b>	<b>20</b>
<b>Urllib .....</b>	<b>21</b>
<b>2. Construcción .....</b>	<b>21</b>

<b>Materiales .....</b>	<b>21</b>
<b>Procedimiento para armar el carrito .....</b>	<b>22</b>
<b>3. Código .....</b>	<b>26</b>
<b>Diagrama de flujo .....</b>	<b>34</b>
<b>Diagrama del circuito.....</b>	<b>35</b>
<b>4. Funcionamiento .....</b>	<b>35</b>
<b>5. Publicación .....</b>	<b>37</b>
<b>GitHub .....</b>	<b>37</b>
<b>Youtube.....</b>	<b>38</b>
<b>Artículo Publicado .....</b>	<b>38</b>
<b>6. Problemas durante el proyecto.....</b>	<b>39</b>
<b>7. Recomendaciones.....</b>	<b>39</b>
<b>8. Conclusiones.....</b>	<b>40</b>
<b>9. Referencias .....</b>	<b>41</b>

### **Presentación**

Este trabajo está elaborado por alumnos de la Escuela Profesional de Ingeniería Informática y de Sistemas de la Universidad Nacional de San Antonio Abad del Cusco, los aportes vertidos aquí, están cimentados en conocimientos previos que se impartieron en la escuela profesional, además de investigación continua y asesoría de amigos, docentes, entre otros. El presente trabajo es parte de un proyecto mucho más grande, cuya finalidad está orientado a los estudiantes y es tener una idea latente de las capacidades de la robótica, conocer las aplicaciones de esta y, de ser posible, aportar con la mejora en el campo de la robótica.

## Introducción

Muchos Actualmente en pleno siglo XXI la disposición de algunas maquinas que constan de hardware y software dependen únicamente de un botón, de un click u otro evento similar, lo cual justifica en cierta medida que la automatización es posible desde el punto de vista científico, pero existen aún factores sociales, morales, éticos, entre otros, que limitan a la automatización mostrar sus máximas capacidades.

Este trabajo como un proyecto de automatización consta de una metodología de investigación en el cual se detalla los mecanismos que se utilizaron para recopilar información referida al tema de automatización de vehículos de prototipo, la funcionalidad de los dispositivos que lleva consigo, y los algoritmos que pueden ser útiles para lograr el objetivo. Adicionalmente de detallan los materiales utilizados cada uno de ellos con sus especificaciones respectivas de modelo, marca y versión.

Seguidamente se explica minuciosamente el desarrollo del proyecto, desde el inicio del recorrido de vehículo pequeño hasta que logra el objetivo, dentro de cuyo proceso se detallan las acciones que indica el algoritmo y las decisiones que se toma de acuerdo al contexto en un tiempo real. Seguido a ello se tienen las recomendaciones que recogen todas las propuestas de parte de los integrantes de este proyecto, las dificultades que hubo y sus consejos correspondientes.

Y finalmente tenemos las conclusiones a las cuales se llegó en el desarrollo de este proyecto, sin duda son bastante alentadores, dado a que tomando como referencia se puede pulir y mejorar aún el proyecto.

## 1. Marco Teórico

### 1.1. Arduino Uno

Arduino es la plataforma de desarrollo de proyectos en electrónica y robótica más utilizada a nivel mundial, esto debido a su facilidad de aprendizaje y uso, abundante documentación y múltiples aplicaciones. Arduino uno R3 es una tarjeta de desarrollo que utiliza el microcontrolador ATmega328P(Atmel), es la versión más recomendada para iniciarse en esta plataforma.

Arduino Uno R3 posee 14 entradas/salidas digitales (6 pueden usarse como PWM), 6 entradas analógicas, un resonador cerámico de 16 MHz, conexión USB, conector de alimentación, conector ICSP y un botón de Reset. La tarjeta contiene todo lo necesario para el funcionamiento del microcontrolador; basta conectarlo al puerto USB o alimentarlo con una fuente de voltaje continuo o una batería para empezar a usarlo.



*Figura 1. Arduino uno*

### ESPECIFICACIONES TÉCNICAS

- Microcontrolador: Atmega328p

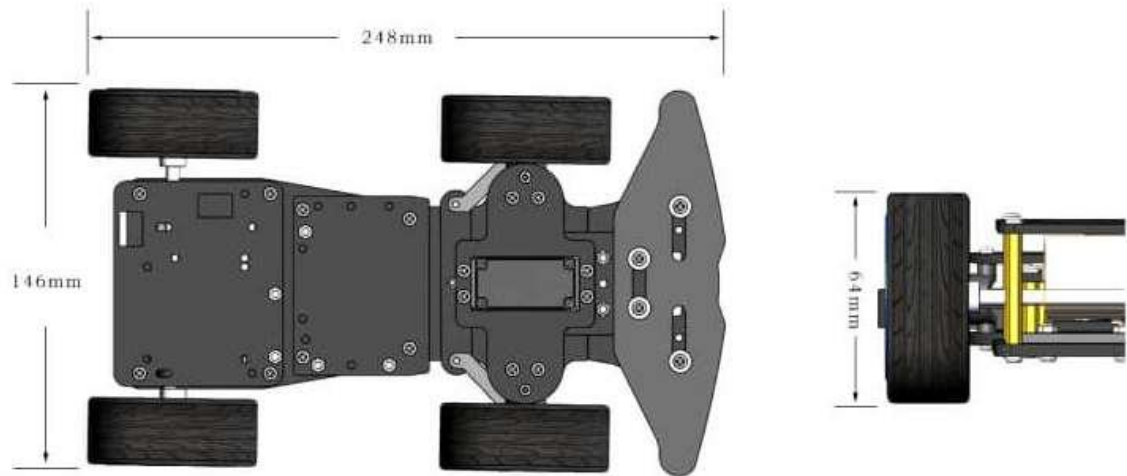
- Chip USB: ATmega16U2
- Voltaje de operación: 5V
- Voltaje de alimentación: 6-20V (/ -12V recomendado)
- Pines digitales I/O: 14 (6 salidas PWM)
- Entradas analógicas: 6
- Corriente máxima entrada/salida: 40mA
- Memoria flash: 32K
- Memoria SRAM: 2k
- Memoria EEPROM: 1k
- Velocidad de reloj: 16Mhz

### 1.2. 4WD Robot Chasis Kit

Y el mando a distancia de control remoto, el uso de dos ruedas a través del manejo servo para controlar la dirección del car. Además, se ha cambiado el mecanismo de manejo de la rueda de reajuste. La configuración de fijación de metal se integra integralmente con el motor y el motor está estrechamente vinculado con la unidad de desplazamiento. El motor adopta el todo-metal de motor de motor. Si usted va de alta velocidad, a lo largo de los 25mm de motor de muñón se puede. El reajuste de la rejilla incorpora dos bridas de las cadenas para garantizar el más posible motor de rotación.

En el caso de que el uso de más precios de las piezas de metal, las cargas, la transmisión de la velocidad, el motor de la rueda, el motor de metal, el manejo de las válvulas, multi-función bracket, etc.), que es caro, el usuario puede comparar los merits, También para el chasis, el trabajo no está en el nivel de los.

Sólo se han fijado los bastidores de montaje para la plataforma, si es que hay otras plataformas, usted necesita punteado o intenta instalar.



*Figura 2. Chasis*

### 1.3. 4 motores DC Gear Motor 1:48

Este motor reductor y llanta es ideal para utilizar la realización de plataformas robóticas. Pensado para construir robots sumo, un robot seguidor de líneas, o cualquier robot con tracción diferencial, este juego de llantas de goma es ideal para darle movimiento a la plataforma robótica. La moto reductora viene con su correspondiente llanta de goma.



*Figura 3. DC Gear Motor 1:48*



**Características:**

- Motor reductor y llanta para aplicaciones de robótica, maquetas y prototipos
- Excelente tracción sobre una gran variedad de superficies
- Llanta con diámetro exterior de 60 mm
- Motor reductor con eje dual (si no hay existencia se envía con eje sencillo)
- Diámetro del eje de salida: 5.3mm aplanado a 3.6
- Voltaje de alimentación 3 a 9 volts
- Corriente de operación nominal (sin carga): 250 mA
- Razón de reducción 1:50

**1.4. Neumático Rueda 65x26mm**

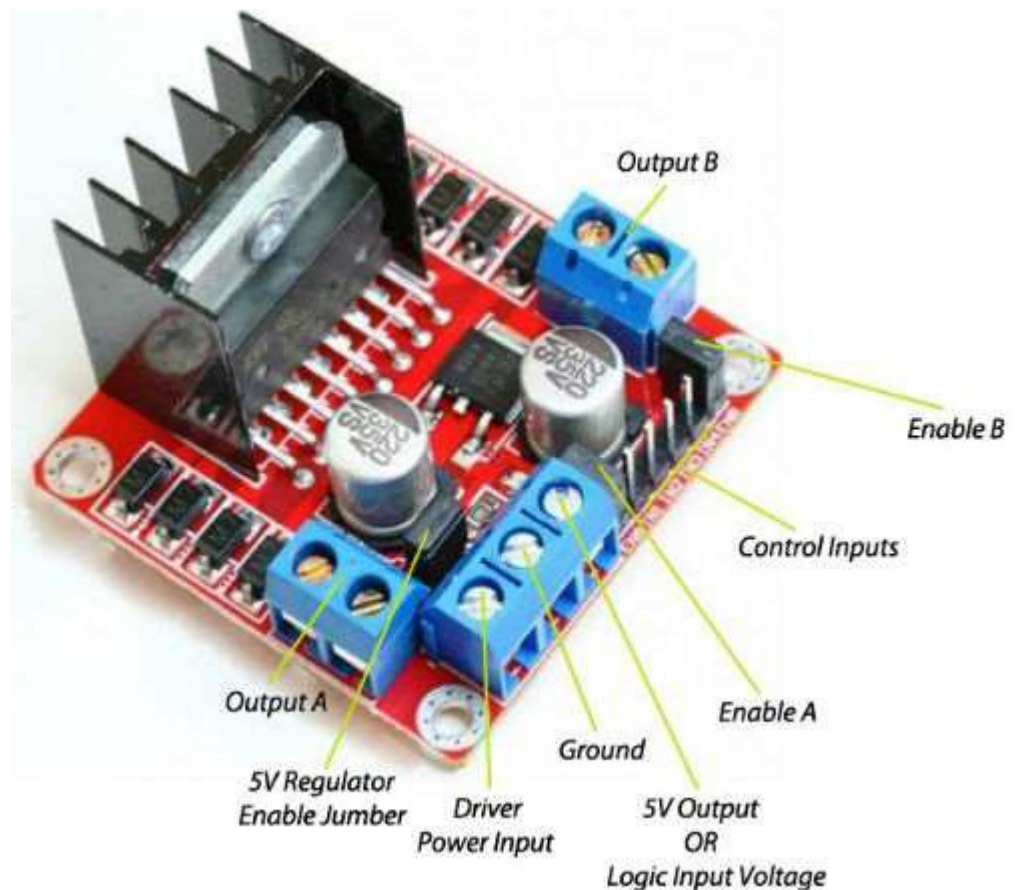
*Figura 4. Neumáticos*

**1.5. Driver Puente H L298N**

El driver L298N es un puente H Dual, NOS permiten controlar 2 motores DC o un motor paso a paso bipolar/unipolar.

El módulo permite controlar el sentido de giro y velocidad mediante señales TTL que se pueden obtener de microcontroladores y tarjetas de desarrollo como Arduino, Raspberry Pi o Launchpads de Texas Instruments.

Tiene integrado un regulador de voltaje de 5V encargado de alimentar la parte lógica del L298N, el uso de este regulador se hace a través de un Jumper y se puede usar para alimentar la etapa de control.



*Figura 5. Driver Puente H L298N*

**Especificaciones técnicas:**

- Chip: L298N
- Canales: 2 (soporta 2 motores DC o 1 motor PAP)
- Voltaje lógico: 5V

- Voltaje de Operación: 5V-35V
- Consumo de corriente (Digital): 0 a 36mA
- Capacidad de corriente: 2A (picos de hasta 3A)
- Potencia máxima: 25W
- Peso: 30g
- Dimensiones: 43 \* 43 \* 27 mm
- Notas
- Voltaje de alimentación, mínimo de 5 V. Posee dos entradas, una de 5V para controlar la parte lógica y otra para alimentar las salidas al motor, que pueden ser de 5V o más.
- Posee un regulador de voltaje de 5V 78M05, para alimentar la etapa lógica del L298N, sin embargo, cuando la alimentación supera los 12V, se recomienda, utilizar una fuente de 5V externa como fuente de alimentación.
- Admite entradas de señal PWM para el control de velocidad.
- Posee 8 diodos de protección contra corriente inversas.

### **Conexión de alimentación**

Este módulo se puede alimentar de 2 maneras gracias al regulador integrado LM7805. Cuando el jumper de selección de 5V se encuentra activo, el módulo permite una alimentación de entre 6V a 12V DC. Como el regulador se encuentra activo, el pin marcado como +5V tendrá un voltaje de 5V DC. Este voltaje se puede usar para alimentar la parte de control del módulo ya sea un microcontrolador o un Arduino, pero recomendamos que el consumo no sea mayor a 500mA.

Cuando el jumper de selección de 5V se encuentra inactivo, el módulo permite una alimentación de entre 12V a 35V DC. Como el regulador no está funcionando, tendremos que conectar el pin de +5V a una tensión de 5V para alimentar la parte lógica del L298N. Usualmente esta tensión es la misma de la parte de control, ya sea un microcontrolador o Arduino.

### 1.6. Batería 18650



Figura 6. Batería 18650

#### Especificaciones:

- Tipo de batería: 18650 Li-ion
- Capacidad: 10800mAh
- Voltaje: 3,7v
- Intensidad de carga: 2A máximo (1A recomendado)
- Circuito de protección: Si, integrado
- Certificaciones: RoHS, CE, UL
- Dimensiones: 70mm x 18.6mm
- Peso: 45,5 gr.

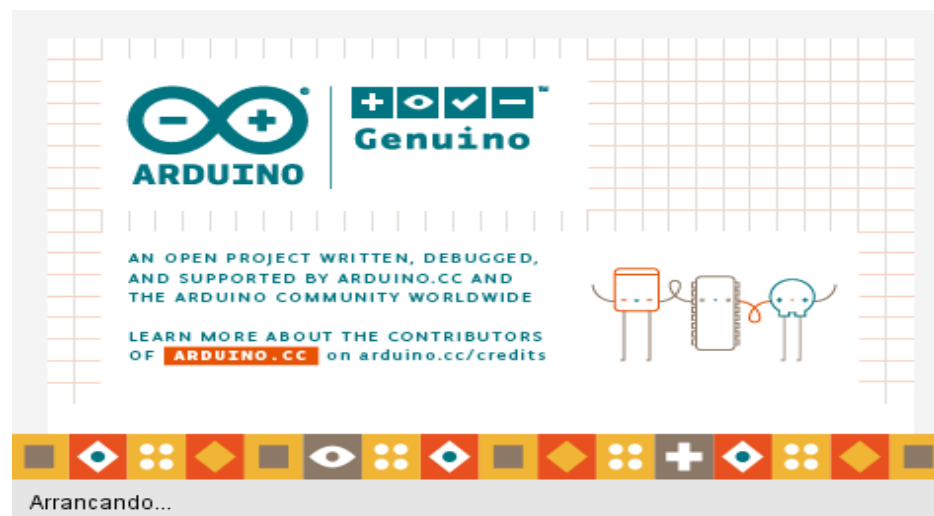
## 1.7. Cables hembra y macho



*Figura 8. Cables*

## 1.8. Arduino 1.8.13

El software consiste en un entorno de desarrollo (IDE) basado en el entorno de processing y lenguaje de programación basado en Wiring, así como en el cargador de arranque (bootloader) que es ejecutado en la placa. El microcontrolador de la placa se programa mediante un computador, usando una comunicación serial mediante un convertidor de niveles RS-232 a TTL serial.



*Figura 9. Arduino 1.8.13*

- **Cuenta con una interfaz de entrada.** Esta puede estar directamente unida a los periféricos, o conectarse a ellos a través de puertos.
- **La interfaz de entrada tiene como objetivo trasladar la información al microcontrolador.** El microcontrolador es la pieza que se encarga de procesar esos datos. Además, varía dependiendo de las necesidades del proyecto en el que se desee usar la placa, y existe una gran variedad de fabricantes y versiones disponibles.
- **También cuenta con interfaz de salida.** Este se encarga de llevar la información procesada a los periféricos autorizados para hacer el uso final de esos datos. En algunos casos puede tratarse de otra placa en la que se centraliza y procesa la información de forma totalmente renovada, o sencillamente, puede ser una pantalla o un altavoz encargado de mostrar la versión final de los datos.

### **Lenguaje de la programación con Arduino: C++**

**¿Qué lenguaje utiliza este tipo de programación?** La plataforma Arduino se programa con un lenguaje propio basado en el lenguaje de programación de alto nivel Processing, lo que significa que es similar a C++.

### **¿Qué quiere decir esto?**

Que se trata de un lenguaje de programación de propósito general asociado a un sistema operativo llamado UNIX. Este lenguaje de medio nivel, trata con objetos básicos como caracteres, números, bits y direcciones de memoria, entre otros.

Este tipo de lenguaje posee una gran portabilidad. Gracias a ello se suele utilizar para la programación de sistemas como la construcción de intérpretes, compiladores, y editores de texto.

Si todavía tienes dudas, te gustaría profundizar en la programación con Arduino y hacer tus primeros pinitos.

### **1.9. Cámara HUAWEI P30 lite**

Cámara trasera:

Cámara de tres lentes: 24 MP (lente gran angular, apertura f/1.8) + 8 MP (lente ultra gran angular) + 2 MP (lente bokeh), admite autofocus.

Sensor de cámara trasera: BSI

Modos de disparo: Lente gran angular, Noche, Retrato, Pro, Cámara lenta, Panorama, Pintar con luz, HDR, Time-Lapse, Panorama 3D, Pegatinas, Documentos, Ultra instantáneas, Capturar sonrisas, Control de audio, Temporizador.

Video Shooting: soporta grabación de video 1080p.

Flash: flash LED trasero único

Disparo continuo: Soportado

Nota:

\*Los píxeles de la foto pueden variar según el modo de disparo.

\*\*Los píxeles de video pueden variar dependiendo del modo de disparo.



*Figura 10. Cámara Huawei P30 lite*

#### **1.10. Bolitas de Plástico de diferentes colores**

Juego de bolas en combinación de colores de moda. Accesorio certificado, hecho de materiales seguros. Bolas son ligeras y muy agradables al tacto, garantizan la funcionalidad.

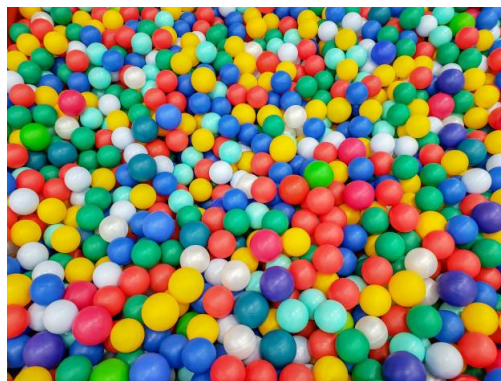
Diámetro de bolas: 7cm.

Las bolas tienen certificado CE, sin BPA.

Tipo(s) de material: Plastico

Composición: 100% polietileno.

Apto para radiocontrol: Exterior e interior



*Figura 11. Bolitas coloridas de plástico*



### 1.11. Python 3.8.6

Lenguaje de programación interpretado, multiparadigma y multiplataforma usado, principalmente, en Big Data, AI (Inteligencia Artificial), Data Science, frameworks de pruebas y desarrollo web. Esto lo convierte en un lenguaje de propósito general de gran nivel debido a su extensa biblioteca, cuya colección ofrece una amplia gama de instalaciones.

Link de descarga Python 3.5.6: <https://www.python.org/downloads/release/python-386/>

### 1.12. Librerías

#### **Numpy**

NumPy es, al igual que Pandas, Matplotlib o Scikit-Learn, uno de los paquetes que no puedes perderte cuando estás aprendiendo Machine Learning, principalmente porque esta librería proporciona una estructura de datos de matriz que tiene algunos beneficios sobre las listas regulares de Python. Algunos de estos beneficios son: ser más compacto, acceder más rápido a leer y escribir artículos, ser más conveniente y más eficiente.

NumPy es un paquete de Python que significa “Numerical Python”, es la librería principal para la informática científica, proporciona potentes estructuras de datos, implementando matrices y matrices multidimensionales. Estas estructuras de datos garantizan cálculos eficientes con matrices.

Link de NUMPY: <https://numpy.org/install/>

#### **Opencv**

OpenCV (Open Source Computer Vision) comenzó como un proyecto de investigación en Intel. Actualmente es la biblioteca de visión por computadora

más grande en términos de funciones poseídas.

Actualmente, OpenCV contiene implementaciones de más de 2500 algoritmos. Además, está disponible de forma gratuita para fines comerciales y académicos. Esta gran biblioteca tiene interfaces para múltiples lenguajes, incluidos Python, Java y C++. La primera versión de OpenCV, 1.0, se lanzó en 2006 y la comunidad de OpenCV ha crecido a pasos agigantados desde entonces.

OpenCV-Python es la API de Python para OpenCV. Además, es compatible con las plataformas más utilizadas, Windows, Mac OS y Linux.

Hay que tener en cuenta que para poder usar esta librería de manera óptima, se deben tener conocimientos en:

- Librería Numpy
- Librería Matplotlib

Para la instalación de OpenCV en Python existen 2 formas:

Desde archivos binarios y archivos fuente precompilados: para ello se debe consultar la documentación pertinente de cada sistema operativo (Por ejemplo Windows y Mac OS. Mediante paquetes para entornos de escritorio estándar (Windows, MacOS y casi cualquier distribución GNU/Linux):

Si solo se necesita módulos principales se debe ejecutar `pip install opencv-python`. Si se necesitan tanto módulos principales como adicionales (contrib) se debe ejecutar `pip install opencv-contrib-python`. Se puede utilizar Jupyter o cualquier IDE de Python para escribir estos comandos.

Link de OpenCV:

[https://docs.opencv.org/master/d5/de5/tutorial\\_py\\_setup\\_in\\_windows.html](https://docs.opencv.org/master/d5/de5/tutorial_py_setup_in_windows.html)

## Serial

Este módulo encapsula el acceso para el puerto serie. Proporciona backends para Python que se ejecuta en Windows, OSX, Linux, BSD (posiblemente cualquier sistema compatible con POSIX) e IronPython. El módulo llamado "serial" selecciona automáticamente el backend apropiado.

Características:

- La misma interfaz basada en clase en todas las plataformas compatibles.
- Acceso a la configuración del puerto a través de las propiedades de Python.
- Soporte para diferentes tamaños de bytes, bits de parada, paridad y control de flujo con RTS / CTS y / o Xon / Xoff.
- Trabajar con o sin recibir tiempo de espera.
- Archivo como API con "leer" y "escribir" ("readline", etc. también es compatible).
- Los archivos de este paquete son 100% Python puro.
- El puerto está configurado para transmisión binaria. Sin eliminación de bytes NULL, traducción CR-LF, etc. (que muchas veces están habilitadas para POSIX). Esto hace que este módulo sea universalmente útil.
- Compatible con iobiblioteca
- Cliente RFC 2217 (experimental), servidor proporcionado en los ejemplos.

Link de Serial: <https://pypi.org/project/pyserial/>

### 1.13. Módulos

#### Time

El módulo `time` de la biblioteca estándar de Python proporciona un conjunto de funciones para trabajar con fechas y/o horas. Además de estas funciones hay otras relacionadas en los módulos `datetime` y `calendar` que conviene conocer.

En el módulo `time` hay funciones para obtener la fecha y/o hora de distintos tipos de relojes (incluido el reloj que ofrece el tiempo civil); para obtener información relacionada con nuestro huso horario; para convertir fechas y/o horas entre distintos tipos; para validar y aplicar formatos de fechas y/o horas y para detener durante un tiempo la ejecución de un programa.

Una fecha y/o hora específica se puede expresar de tres modos diferentes: como un número (`float`), como una cadena y como un objeto `struct_time` (es una tupla a medida (`namedtuple`) con nueve valores enteros).

Por ejemplo, la fecha-hora 4-3-2017 19:10:01 representada en los tipos anteriores:

como un número: 1488651001.7188754 segundos como una cadena: "Sat Mar 4 19:10:01 2017" y como un objeto `struct_time`: `time.struct_time(tm_year=2017, tm_mon=3, tm_mday=4, tm_hour=18, tm_min=10, tm_sec=1, tm_wday=5, tm_yday=63, tm_isdst=0)`

Algunas funciones del módulo `time` operan sólo en un rango temporal denominado época que comienza el día 1 de enero de 1970 a las 0 horas y que finaliza en el año 2038 (en un sistema de 32 bit).

El tiempo civil se puede expresar como tiempo local, o bien, como tiempo UTC que es el Tiempo Universal Coordinado para/por todos los países. La diferencia

entre ambos tiempos depende del punto geográfico (que se corresponde con un huso horario de 24 existentes) y de los ajustes que algunos países aplican en el horario de verano.

### Urllib

El El módulo Urllib es el módulo de manejo de URL para Python. Se utiliza para buscar URL (localizadores uniformes de recursos). Utiliza la función urlopen y es capaz de buscar URL utilizando una variedad de protocolos diferentes.

Urllib es un paquete que recopila varios módulos para trabajar con URL, como:

- urllib.request para abrir y leer.
- urllib.parse para analizar URL
- urllib.error para las excepciones planteadas
- urllib.robotparser para analizar archivos robot.txt

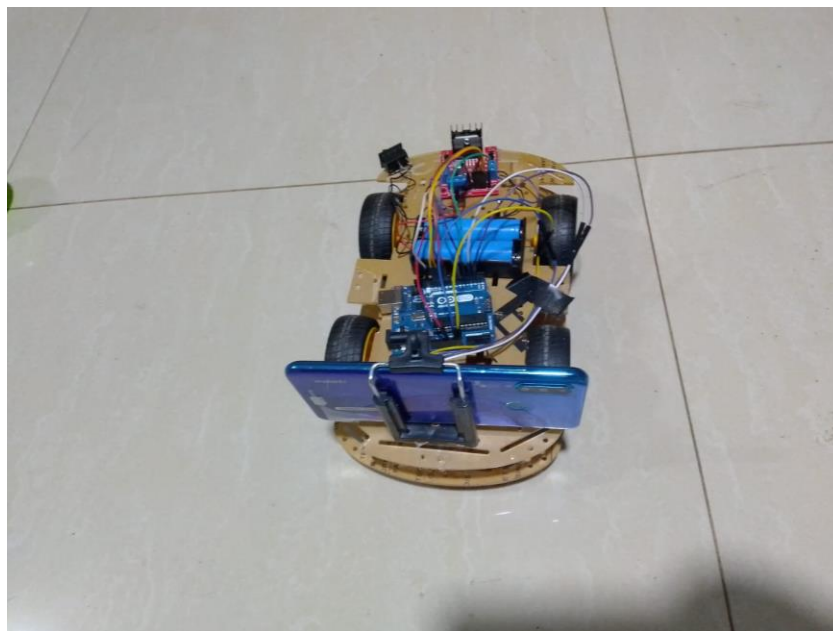
## 2. Construcción

### Materiales

Nro.	Material	Cantidad	Costo
1	Chasis de plástico para el soporte del carrito con 4 ruedas.	1	57 soles
2	Módulo de Control L298N	1	14.7 soles
3	Motores DC (7 – 12 voltios)	4	5 soles
4	Arduino UNO V3 Atmega 328	1	25 soles
5	Bluetooth hc-05	1	33 soles
6	Baterías 2800mAh (3.7V)	2	26 soles
7	Cables	6	1.6 soles
8	Celular Smartphone Huawei P30 lite.	1	865 soles
9	App iriumcam	1	0 soles
10	Sensor HC-SR04	1	9 soles
11	Soporte para el celular	1	20 soles
12	Pelotas esféricas de colores.	1	8 soles

**Procedimiento para armar el carrito**

- Se colocan las ruedas en el chasis con la estructura básica de un carro.
- Se conectan cada rueda a los cuatro motores.
- Conecta los motores al módulo de control L298N.
- Se conectan los 6 puertos de salida del módulo L298N al arduino uno.
- Se alimenta el módulo L298N con las 2 baterías en serie cada una de 3.5.
- El arduino recibe el voltaje de 5V por la regulación del módulo L298N y asigna la tierra GND.
- Se quema el código creado al Arduino uno para su posterior uso.
- Una vez teniendo la construcción del carrito agregamos el soporte de celular para servir de soporte al celular HUAWEI P30 lite.



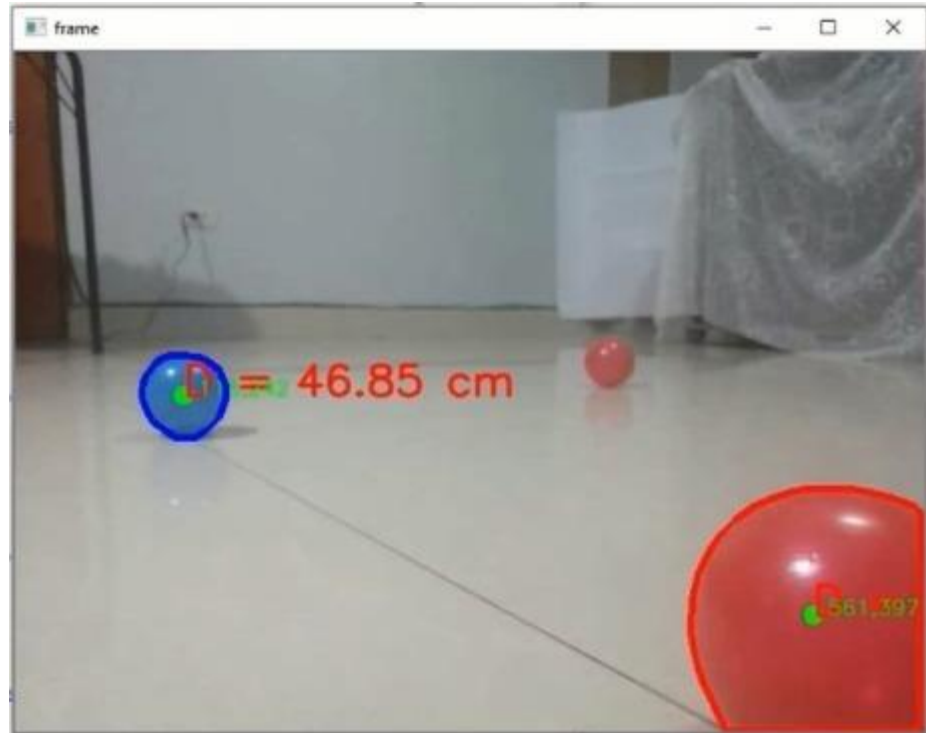
*Figura 12. Armado del carrito*



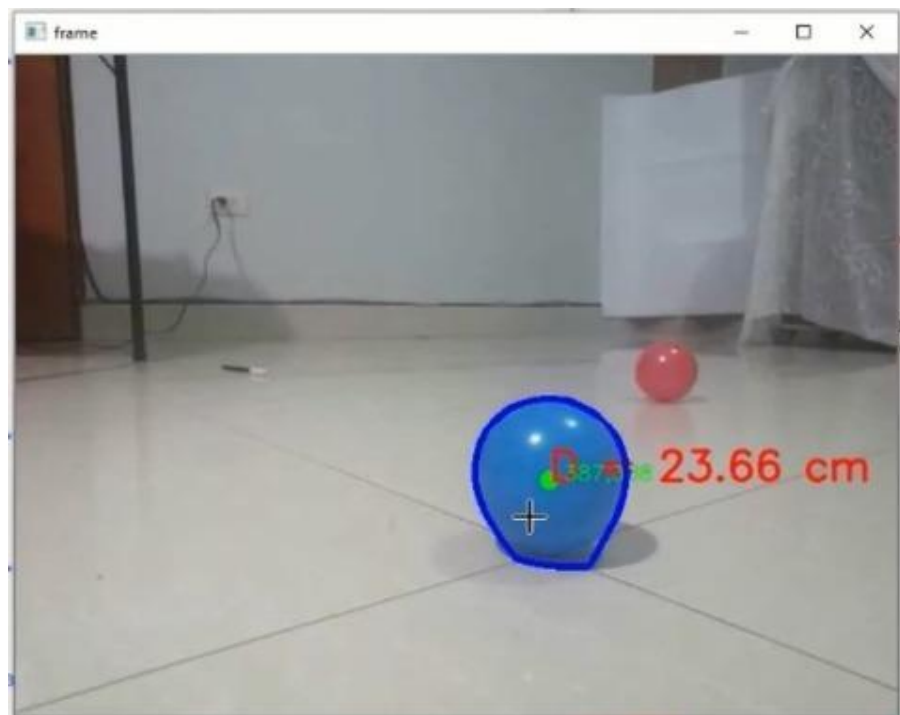
*Figura 13.* Armado del chasis con las bolas de plástico



*Figura 14.* Armado del chasis con las bolas de plástico



*Figura 15.* Reconocimiento de distancia del carrito con la cámara del celular.



*Figura 16.* Reconocimiento de distancia.



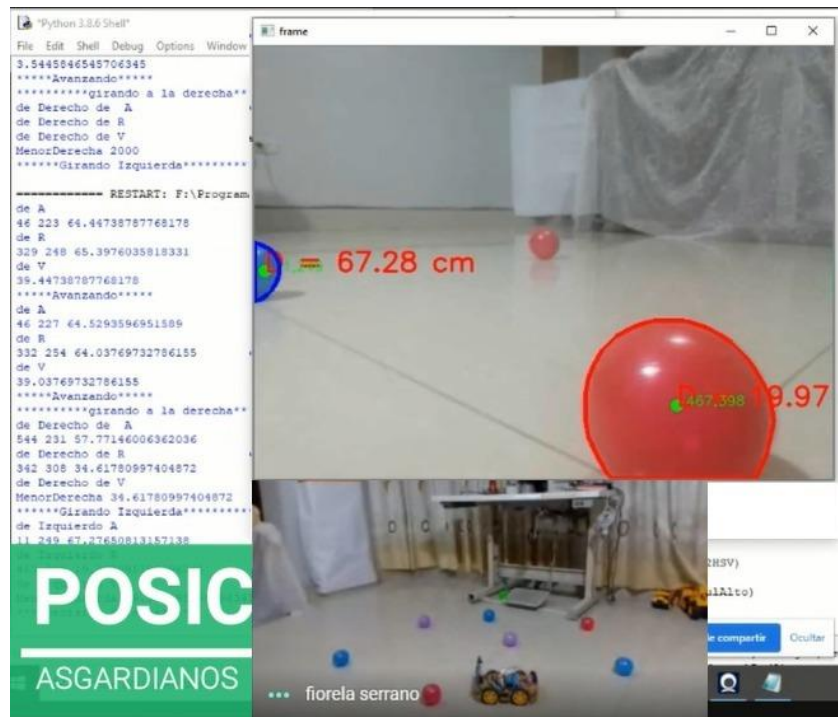


Figura 17. Ejecución del Proyecto.

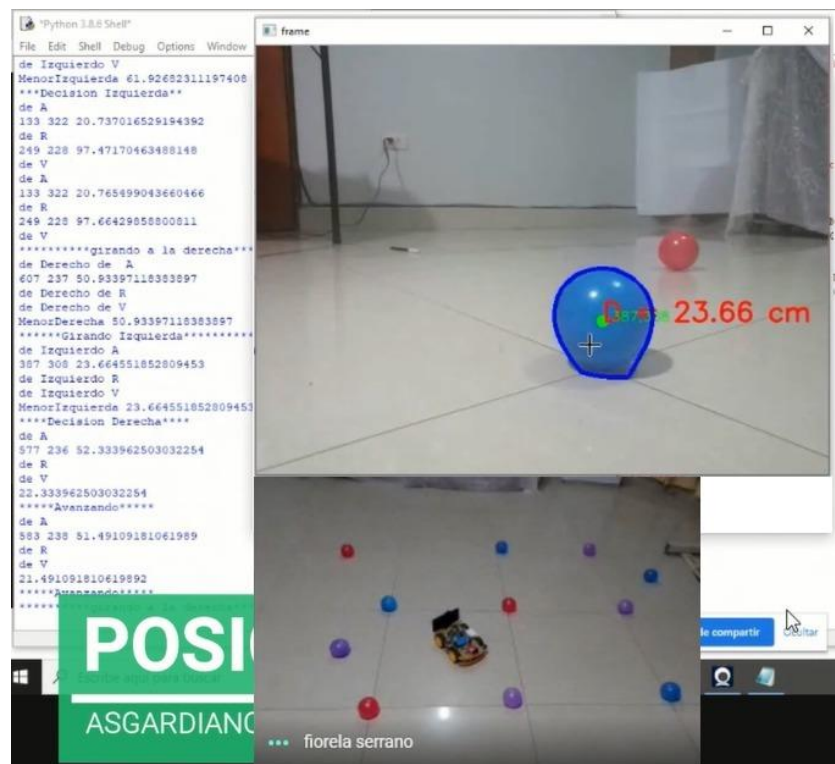


Figura 18. Ejecución del Proyecto, distancia y posicionamiento.

### 3. Código

```

*****
***** CARRITO POR BLUETOOTH *****
*****
# Asignatura: Robotica y Procesamiento de Señal
# Semestre: 2020-I
*****
*****

#===== LIBRERIAS =====
#Importamos la librerias
import cv2
import numpy as np
import urllib
import urllib.request
import math
import serial
import time
#Importamos la libreria serial para conectar Arduino con Python

#===== PROGRAMA =====
# Iniciamos la comunicacion con el arduino con el puerto COM4
# Y una comunicacion serial de 9600
ser =serial.Serial('COM4',9600)
#definimos un procedimiento para dibujar contornos y hallar coordenadas
def dibujar(mask,color,w,f):
    contornos,_ = cv2.findContours(mask, cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    #Se recorre cada uno de los contornos encontrados
    IN=[]
    for c in contornos:
        #Determinamos el area en pixiles del contorno
        area = cv2.contourArea(c)
        #Si el area es mayor al valor se dibuja
        A=[]
        if area > 1000:
            r=math.sqrt(area/math.pi)
            p=r*2
            distancia = (w*f)/p
            #buscamos areas centrales
            M = cv2.moments(c)
            #un if por si el denominador es 0 y pueda ser indeterminado

```

```

    if (M["m00"]==0): M["m00"]=1
    #Encontramos coordenadas X y Y
    x = int(M["m10"]/M["m00"])
    y = int(M['m01']/M['m00'])
    A.append(x)
    A.append(y)
    A.append(distancia)
    IN.append(A)
    #mejora la visualizacion del contorno
    nuevoContorno = cv2.convexHull(c)
    #dibujamos un circulo de 7 pixeles en el centro del objeto encontrado XY
    cv2.circle(image,(x,y),7,(0,255,0),-1)
    #para visualizar el texto
    mensaje="D = "+str(round(distancia,2))+ " cm"
    cv2.putText(image,mensaje,(x,y),font,1,(0,0,255),2,cv2.LINE_AA)
    cv2.putText(image,'{},{}'.format(x,y),(x+10,y), font,
0.50,(0,255,0),1,cv2.LINE_AA)
    #dibujamos los contornos
    cv2.drawContours(image, [nuevoContorno], 0, color, 3)
    return IN

##Determinar la distancia Menor de un arreglo
def DistanciaMArreglo(A):

    if (len(A)==0):
        return 2000;
    else :
        menor=A[0][2]
        for k in range(1, len(A)):
            if(menor>A[k][2]):
                menor=A[k][2]
        return menor

##Modulo para determinar la distancia menor
def DistanciaMTotal(A,B,C):
    menor1=DistanciaMArreglo(A)
    menor2=DistanciaMArreglo(B)
    menor3=DistanciaMArreglo(C)

    menor=menor1
    if(menor>menor2):
        menor=menor2
    if(menor>menor3):

```

```
        menor=menor3
    if(menor>menor3):
        menor=menor3
    return menor

def EscribirMatriz(A):
    for k in range(len(A)):
        print(A[k][0],A[k][1],A[k][2])

#Creamos una variable de conexion IP con la camara
url = 'http://192.168.1.3:8080/shot.jpg'

#Definimos el rango de valores del azul
azulBajo = np.array([100,100,20],np.uint8)
azulAlto = np.array([125,255,255],np.uint8)

#Definimos el rango de valores del rojo
redBajo1 = np.array([0,100,20],np.uint8)
redAlto1 = np.array([5,255,255],np.uint8)

redBajo2 = np.array([175,100,20],np.uint8)
redAlto2 = np.array([179,255,255],np.uint8)

#Definimos el rango de valores del Violeta
violetaBajo = np.array([140, 50, 20], np.uint8)
violetaAlto = np.array([170, 255, 255], np.uint8)
#Definimos el rango de valores del Verde
verdeBajo = np.array([36, 50, 20], np.uint8)
verdeAlto = np.array([75, 255, 255], np.uint8)

font = cv2.FONT_HERSHEY_SIMPLEX
disP=30
#Iniciamos el programa en un bucle
while True:
    ser.write(str('5').encode())

    #Iniciamos la transmision de la camara
    imgResp = urllib.request.urlopen(url)
    imgNp = np.array(bytearray(imgResp.read()),dtype = np.uint8)
    #Almacenamos la transmision en una variable image
```

```

image = cv2.imdecode(imgNp,-1)

#Transformamos la imagen de BGR a HSV
frameHSV = cv2.cvtColor(image,cv2.COLOR_BGR2HSV)
#Obtenemos una imagen binaria del azul
maskAzul = cv2.inRange(frameHSV,azulBajo,azulAlto)

#Obtenemos una imagen binaria del rojo
maskRed1 = cv2.inRange(frameHSV,redBajo1,redAlto1)
maskRed2 = cv2.inRange(frameHSV,redBajo2,redAlto2)
maskRed = cv2.add(maskRed1,maskRed2)

#Obtenemos una imagen binaria del violeta
maskVerde = cv2.inRange(frameHSV,verdeBajo,verdeAlto)

#Dibujamos los contornos encontrados segun su color
time.sleep(2)
A=dibujar(maskAzul,(255,0,0),6,450)
R=dibujar(maskRed,(0,0,255),6,653)
V=dibujar(maskVerde,(0,255,0),6,653)
cv2.imshow('frame',image)
print("de A")
EscribirMatriz(A)
print("de R")
EscribirMatriz(R)
print("de V")
EscribirMatriz(V)
menor=DistanciaMTotal(A,R,V)
if(menor==2000):
    for k in range (int(15)):
        ser.write(str('1').encode())
        ser.write(str('5').encode())

if(menor>disP and menor<1000):
    dM=menor-disP
    print(dM)
    print("*****Avanzando*****")
    for k in range (int(dM)):
        ser.write(str('1').encode())
        ser.write(str('5').encode())

#Iniciamos la transmision de la camara
imgResp = urllib.request.urlopen(url)

```

```

imgNp = np.array(bytearray(imgResp.read()),dtype = np.uint8)
#Almacenamos la transmision en una variable image
image = cv2.imdecode(imgNp,-1)

#Transformamos la imagen de BGR a HSV
frameHSV = cv2.cvtColor(image,cv2.COLOR_BGR2HSV)
#Obtenemos una imagen binaria del azul
maskAzul = cv2.inRange(frameHSV,azulBajo,azulAlto)

#Obtenemos una imagen binaria del rojo
maskRed1 = cv2.inRange(frameHSV,redBajo1,redAlto1)
maskRed2 = cv2.inRange(frameHSV,redBajo2,redAlto2)
maskRed = cv2.add(maskRed1,maskRed2)

#Obtenemos una imagen binaria del violeta
maskVerde = cv2.inRange(frameHSV,verdeBajo,verdeAlto)

#Dibujamos los contornos encontrados segun su color
time.sleep(2)
A=dibujar(maskAzul,(255,0,0),6,450)
R=dibujar(maskRed,(0,0,255),6,653)
V=dibujar(maskVerde,(0,255,0),6,653)
cv2.imshow('frame',image)
print("de A")
EscribirMatriz(A)
print("de R")
EscribirMatriz(R)
print("de V")
EscribirMatriz(V)
menor=DistanciaMTotal(A,R,V)
if(menor==2000):
    for k in range (int(15)):
        ser.write(str('1').encode())
        ser.write(str('5').encode())
if(menor>disP and menor<1000):
    dM=menor-disP
    print(dM)
    print("*****Avanzando*****")
    for k in range (int(dM)):
        ser.write(str('1').encode())
        ser.write(str('5').encode())
#####Girar a la derecha y captar imagen #####
print("*****girando a la derecha*****")

```

```

for k in range (5):
    ser.write(str('7').encode())
    ser.write(str('5').encode())
time.sleep(2)
#Iniciamos la transmision de la camara
imgResp = urllib.request.urlopen(url)
imgNp = np.array(bytearray(imgResp.read()),dtype = np.uint8)
#Almacenamos la transmision en una variable image
image = cv2.imdecode(imgNp,-1)
#Transformamos la imagen de BGR a HSV
frameHSV = cv2.cvtColor(image,cv2.COLOR_BGR2HSV)
#Obtenemos una imagen binaria del azul
maskAzul = cv2.inRange(frameHSV,azulBajo,azulAlto)

#Obtenemos una imagen binaria del rojo
maskRed1 = cv2.inRange(frameHSV,redBajo1,redAlto1)
maskRed2 = cv2.inRange(frameHSV,redBajo2,redAlto2)
maskRed = cv2.add(maskRed1,maskRed2)

#Obtenemos una imagen binaria del violeta
maskVerde = cv2.inRange(frameHSV,verdeBajo,verdeAlto)
time.sleep(2)
A=dibujar(maskAzul,(255,0,0),6,450)
R=dibujar(maskRed,(0,0,255),6,653)
V=dibujar(maskVerde,(0,255,0),6,653)
print("de Derecho de A")
EscribirMatriz(A)
print("de Derecho de R")
EscribirMatriz(R)
print("de Derecho de V")
EscribirMatriz(V)
cv2.imshow('frame',image)
DmDe=DistanciaMTotal(A,R,V)
print("MenorDerecha",DmDe)

#####Girar a la izqueirda y captar imagen #####
print("*****Girando Izquierda*****")
for k in range (10):
    ser.write(str('6').encode())
    ser.write(str('5').encode())
time.sleep(2)
#Iniciamos la transmision de la camara

```

```

imgResp = urllib.request.urlopen(url)
imgNp = np.array(bytearray(imgResp.read()),dtype = np.uint8)
#Almacenamos la transmision en una variable image
image = cv2.imdecode(imgNp,-1)
#Transformamos la imagen de BGR a HSV
frameHSV = cv2.cvtColor(image,cv2.COLOR_BGR2HSV)
#Obtenemos una imagen binaria del azul
maskAzul = cv2.inRange(frameHSV,azulBajo,azulAlto)

#Obtenemos una imagen binaria del rojo
maskRed1 = cv2.inRange(frameHSV,redBajo1,redAlto1)
maskRed2 = cv2.inRange(frameHSV,redBajo2,redAlto2)
maskRed = cv2.add(maskRed1,maskRed2)

#Obtenemos una imagen binaria del violeta
maskVerde= cv2.inRange(frameHSV,verdeBajo,verdeAlto)
time.sleep(2)
A=dibujar(maskAzul,(255,0,0),6,450)
R=dibujar(maskRed,(0,0,255),6,653)
V=dibujar(maskVerde,(0,255,0),6,653)
print("de Izquierdo A")
EscribirMatriz(A)
print("de Izquierdo R")
EscribirMatriz(R)
print("de Izquierdo V")
EscribirMatriz(V)
cv2.imshow('frame',image)
DmIz=DistanciaMTotal(A,R,V)
print("MenorIzquierda",DmIz)

#####
if(DmDe>DmIz):
    print("****Decision Derecha****")
    for k in range(10):
        ser.write(str('7').encode())
        ser.write(str('5').encode())
        time.sleep(2)
else:
    print("****Decision Izquierda****")

```



```
#dibujar(maskRosa,(255,0,255))

#mostramos el video
cv2.imshow('frame',image)
#Rango valores 0-213 213-426 426-640
if cv2.waitKey(1) and 0xFF == ord('s'):
    break

#Finalmente cerramos todas las ventanas que se hubieran podido abrirse
cv2.destroyAllWindows()
```

## Diagrama de flujo

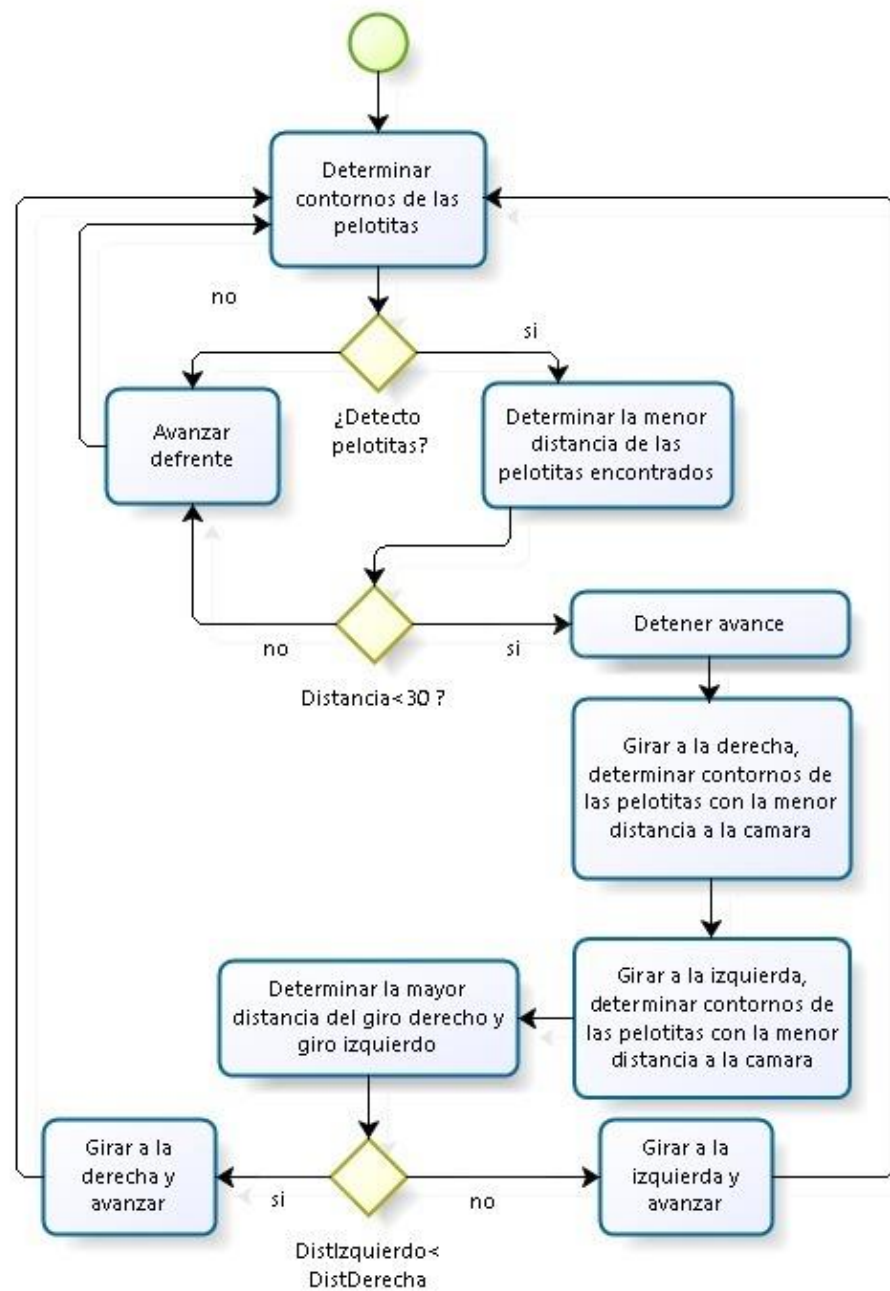


Figura 19. Diagrama de flujo

## Diagrama del circuito

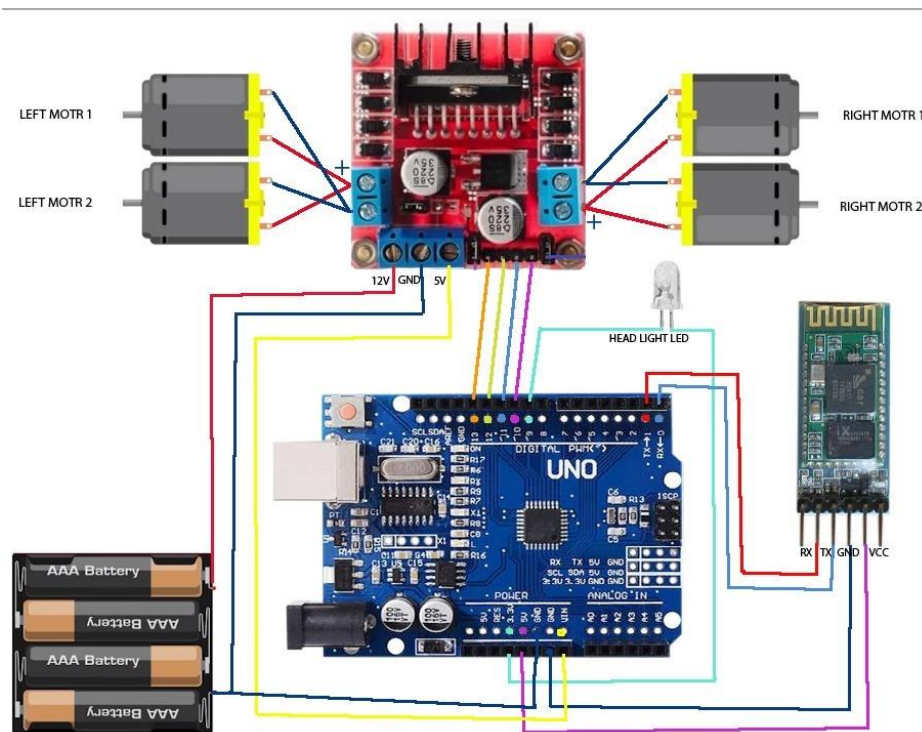


Figura 20. Diagrama de circuito

## 4. Funcionamiento

Antes de ejecutar el programa se colocan pelotas esféricas de tres colores en una disposición aleatoria, estos hacen el papel de obstáculos durante el recorrido del vehículo pequeño de prototipo, luego se compila el programa, las acciones de comunicación entre el Arduino Uno y el ordenador dan inicio a través de dispositivo Bluetooth, y posterior a ello no es necesario intervenir en el programa, la ejecución se realiza automáticamente hasta lograr su objetivo.

El vehículo pequeño tiene la versatilidad de poder empezar en cualquier posición respecto al conjunto de obstáculos (pelotitas pequeñas), las primeras acciones que realiza son el mapeo de panorama delante suyo a través del celular que se comunica utilizando http y una dirección ip, dentro de este mapeo se clasifican los obstáculos por el tipo de

color con la ayuda del rango de colores en un modelo de color HSV(Hue, Saturation, Value); cada conjunto de un determinado color se lista dentro de un arreglo donde se guarda la distancia de la cámara al obstáculo, de cada uno de estos grupos de colores es seleccionada la distancia menor, obteniendo así tantas distancias menores como grupos de colores mapeados hubo, posterior a ello se elige la distancia general menor a partir de la lista anterior.

La longitud de acercamiento se determina simplemente realizando una resta de la distancia general menor menos 30 cm, con lo cual garantizamos que la siguiente acción que vaya a tomar estará cerca de la periferia de un obstáculo antes de chocar, en este punto de decisión, primero se mueve hacia la derecha en un promedio de 20°, mapea el panorama que tiene en frente y en base a ello calcula la distancia de la cámara al obstáculo más próximo, luego gira 20° hacia la izquierda respecto a la posición de decisión, análogamente, mapea el panorama y calcula la distancia de la cámara al obstáculo más cercano, la decisión que toma será avanzar al lado cuya distancia sea el mayor, dado que permite al vehículo un contexto de movimiento cómodo y con pocas complicaciones.

El proceso se realiza de manera repetitiva hasta que logre el objetivo de avanzar desde un punto hasta un extremo sin chocar con los obstáculos.

## 5. Publicación

### GitHub

FernandoCallasaca / CARRITO-ESQUIVANDO-PELOTITAS-CON-ARDUINO-Y-OPENCV

Unwatch 1 Star 0 Fork 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

FernandoCallasaca Update README.md 1cbb3f 3 minutes ago 6 commits

Images	Add files via upload	8 minutes ago
Esquivando_Pelotitas.py	Add files via upload	35 minutes ago
README.md	Update README.md	3 minutes ago

About

Carrito esquivando pelotitas con arduino y opencv.

Readme

Releases

No releases published

366 lines (298 sloc) 11.6 KB

Raw Blame

## CARRITO-ESQUIVANDO-PELOTITAS-CON-ARDUINO-Y-OPENCV

Universidad Nacional de San Antonio Abad del Cusco (UNSAAC)

Departamento Académico de Informática

Robótica y Procesamiento de Señales

Integrantes

- BOLAÑOS CCOPA, RONY - 154624
- CALLASACA ACUÑA, FERNANDO - 140989
- LIMPE QUISPE, JERRY ANDERSON - 140985
- MENDOZA CUIRO, NILO FIDEL - 144987
- SERRANO AMAU, FIORELA ELOISA - 133968

Presentación

Este trabajo está elaborado por alumnos de la Escuela Profesional de Ingeniería Informática y de Sistemas de la Universidad Nacional de San Antonio Abad del Cusco, los aportes vertidos aquí, están cimentados en conocimientos previos que se impartieron en la escuela profesional, además de investigación continua y asesoría de amigos, docentes, entre otros. El presente trabajo es parte de un proyecto mucho más grande, cuya finalidad está orientado a los estudiantes y es tener una idea latente de las capacidades de la robótica, conocer las aplicaciones de esta y, de ser posible, aportar con la mejora en el campo de la robótica.

### Código comentado

```
#####
##### CARRITO POR BLUETOOTH #####
#####
# Asignatura: Robotica y Procesamiento de Señal
# Semestre: 2020-I
#####

##### LIBRERIAS #####
#Importamos la librerias
import cv2
import numpy as np
import urllib
import urllib.request
import math
import serial
import time
#Importamos la libreria serial para conectar Arduino con Python
```

Carrito en el circuito de pelotitas:



Construido por nuestra compañera Fiorela.

*Figura 21.* Publicación en GitHub

Link:

<https://github.com/FernandoCallasaca/CARRITO-ESQUIVANDO-PELOTITAS-CON-ARDUINO-Y-OPENCV>

## Youtube

Link:

<https://youtu.be/c9A1PSKTCsQ>

## Artículo Publicado

Link:

[https://www.researchgate.net/publication/345260025\\_CARRITO\\_PROTOTIPO\\_ESQUIVANDO\\_OBSTACULOS\\_ESFERICOS\\_CON\\_ARDUINO\\_UNO\\_Y\\_OPEN\\_CV](https://www.researchgate.net/publication/345260025_CARRITO_PROTOTIPO_ESQUIVANDO_OBSTACULOS_ESFERICOS_CON_ARDUINO_UNO_Y_OPEN_CV)

## 6. Problemas durante el proyecto

- Problemas para determinar distancias.
- Dificultades al momento de reconocer esferas y así poder trabajar de manera más precisa.
- Problemas con el rango y la calibración de los colores.
- Dificultad al tratar de reconocer 7 colores, por ende, decidimos trabajar con 3 colores (rojo, verde y azul).

## 7. Recomendaciones

- Para obtener un rendimiento más óptimo, se recomienda trabajar la imagen a una escala de grises, es decir convertir una imagen real a una imagen binaria a blanco y negro y sobre ella obtener los contornos de dicha imagen.
- Para obtener un mayor rendimiento en el procesamiento de las imágenes es recomendable mantener al equipo que recibe la información que esté más cerca al que envía la información.
- Recomendamos trabajar en ambientes donde no obtengamos mucha luz porque los colores tienden a reflejarse en mayor proporción.
- Saber cómo funcionan las cosas nos ayudará a mantener la calma y saber qué plan de acción tomar.

## 8. Conclusiones

En conclusión, el proyecto desarrollado en su estructura básica pone en evidencia que la automatización de máquinas es posible, viable y cada vez superable. Este proyecto a escala mayor dotado de algunas funcionalidades más, se podría utilizar en campos como la exploración espacial de suelos desconocidos, justamente por tratarse de un escenario incierto.



## 9. Referencias

- Palleta, S. (2019, 27 marzo). Circuit design Servo motor controlado mediante sensor ultrasonico(HC-SR04) 2.0. Tinkercad.  
<https://www.tinkercad.com/things/0CmtapmAscY-servo-motor-controlado-mediante-sensor-ultrasonicohc-sr04-20>
- Naylamp, N. M. (2016, 13 abril). Tutorial de Arduino y sensor ultrasónico HC-SR04. Naymalp emchatronics. [https://naylampmechatronics.com/blog/10\\_Tutorial-de-Arduino-y-sensor-ultras%C3%B3nico-HC-S.html](https://naylampmechatronics.com/blog/10_Tutorial-de-Arduino-y-sensor-ultras%C3%B3nico-HC-S.html)
- S.L. (2014, 27 abril). Arduino UNO + Servo + HC-SR04. SPAINLABS.  
<https://www.spainlabs.com/foros/tema-Arduino-UNO-Servo-HC-SR04>
- Ciril, C. (2018, 16 junio). Activar servo giro continuo con sensor HC SR04. Arduino CC.  
<https://forum.arduino.cc/index.php?topic=407373.0>
- teckel12 / Arduino New Ping / Downloads. (2018, 15 julio). Bitbucket.
- Normas APA. Formato APA para la presentación de trabajos escritos.  
<https://normasapa.com/formato-apa-presentacion-trabajos-escritos/>
- Jose Mauro Pillco(2020). Trabajos y proyectos de semestres anteriores.  
<https://classroom.google.com/c/ODg4NTgyNDk5MzZa/m/MTM4NDk1MTk3NzYx/details>