

# Robot verificador de temperatura y desinfectador por Covid 19

UNIVERSIDAD RICARDO PALMA  
ESCUELA INGENIERIA MECATRONICA

Nombres de Autores: Paucar Coaquira Luis Fernando,  
Donayre Maraion Oscar Domingo,  
Profesor : Dr. Ing. Ricardo J. Palomares Orihuela

## I. INTRODUCCIÓN

Debido a la presencia de la pandemia COVID y el aumento de casos, las probabilidades de contagio son muy altas. Los síntomas que esta puede ocasionar pueden complicarse hasta ser mortales. Aunque actualmente se está distribuyendo la cura, el proceso de vacunación es lento, por lo que aún se sigue estableciendo la cuarentena y el límite reducido de personas en cada área local y aforo.

La exposición frecuente que tiene que realizar el personal encargado de revisar la temperatura hace que arriesgue su salud y sea muy probable que contraiga el virus, haciendo un portador adicional de la pandemia. Este tipo de trabajo hace que se ocupe un puesto de trabajo innecesariamente cuando el personal puede estar laborando en trabajos más útiles.

Como solución para disminuir la tasa de contagios y seguir propagando el virus, hemos propuesto como solución un robot estacionario vigía, que se encargará de tomar las temperaturas de cada persona que va a ingresar a una sucursal, permitiendo y prohibiendo su ingreso de acuerdo a la temperatura registrada.

## II. DISEÑO MECÁNICO

El robot es un poliarticulado, que girará inicialmente por su base y que contará con dos brazos, uno para la medición de temperatura y el otro para el rociamiento de alcohol farmacéutico.



Figure 1. Modelo Robótico Final en SolidWork

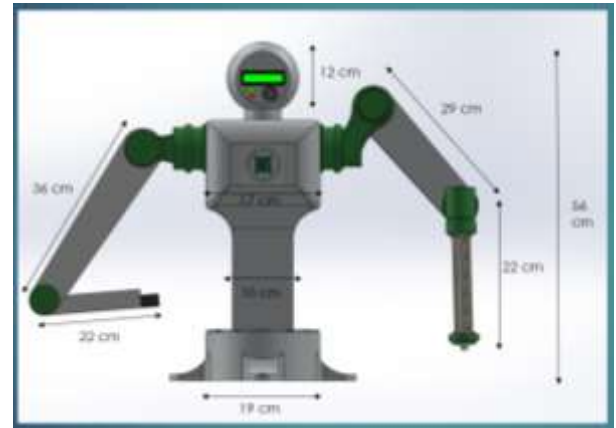


Figure 2. Medida de los Eslabones

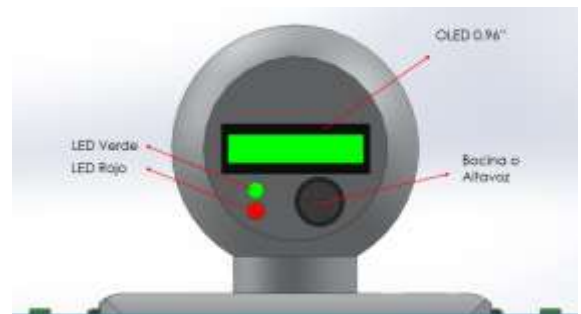


Figure 3. Componentes indicadores

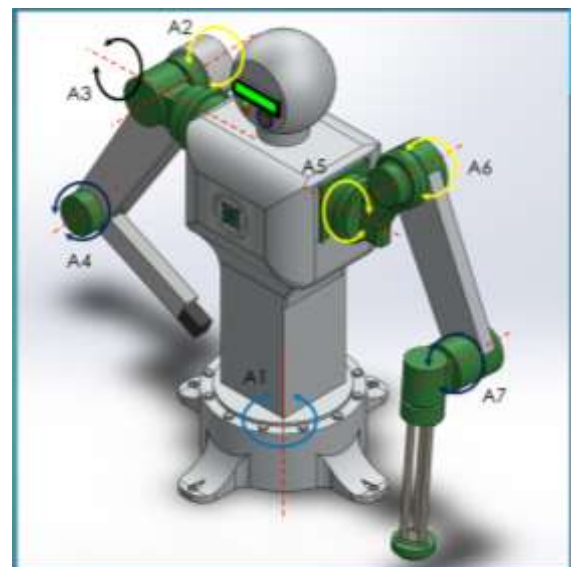


Figure 4. Sentido de giro de las articulaciones

### III. DISEÑO ELECTRICO

Para nuestro diseño eléctrico implementamos el sistema Arduino para la conexión lógica junto con los sensores. Y en cuanto a la motricidad y la activación de mecanismos se implementó también servomotores con alimentación externa.

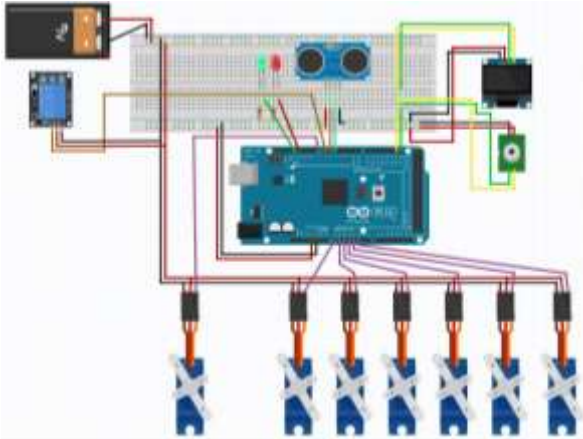


Figure 5. Esquema Electrónico del proyecto en Fritzing

### IV. CODIGO DE PROGRAMACION

```

Proyecto_Robotico_Alternativo Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda

Proyecto_Robotico_Alternativo
#include <Servo.h>
#include <mlx90615.h> // Debe descargar la libreria que conecta al sensor MLX90614
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

//Adafruit_MLX90614 mlx = Adafruit_MLX90614();
MLX90615 mlx = MLX90615();
int alarma = 30; // valor de temperatura de alarma

//Definiendo los pines OLED y su fuente de letra
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
// The pins for I2C are defined by the Wire-library.
// On an Arduino Uno: AA(SDA), A5(SCL)
// On an Arduino Mega 2560: 20(SDA), 21(SCL)
// On an Arduino Leonardo: 2(SDA), 3(SCL), ...
#define OLED_RESET 4 // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_ADDRESS 0x3C // See datasheet for Address: 0x3D for 128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, 4);

const int trigger=8, echo=9, rojo=12, verde=13, rociador=7;
long tiempo;
double distancia;
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
Servo servo6;
Servo servo7;

void setup() {
  pinMode(trigger, OUTPUT);
  pinMode(echo, INPUT);
  Serial.begin(9600);
  Serial.println("Molex MLX90615 infra-red temperature sensor test");
  mlx.begin();
  Serial.println("Sensor ID number = ");
  Serial.println(mlx.get_id(), HEX);
}

```

Figure 6. Programa en Arduino parte 1

```

//For OLED I2C
if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
  Serial.println(F("SSD1306 allocation failed"));
  for(;;) // Don't proceed, loop forever
}
display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS);
delay(2000);
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);

// Declaramos los LEDs de Temperatura
pinMode(rojo, OUTPUT);
pinMode(verde, OUTPUT);
//Rojo = alta temperatura - Verde = Temperatura Normal
digitalWrite(rojo, LOW);
digitalWrite(verde, LOW);

//Declaramos el pin del rociador:
pinMode(rociador, OUTPUT);
digitalWrite(rociador, LOW);

//Iniciamos el servomotor Principal para el pin 8
servo1.attach(8);

//Declaramos los demas servomotores a otros pines:
servo2.attach(A0);
servo3.attach(A1);
servo4.attach(A2);
servo5.attach(A3);
servo6.attach(A4);
servo7.attach(A5);

Serial.println("Prueba Temperatura");

```

Figure 7. Programa en Arduino parte 2

```

void loop() {
  float t = mlx.get_object_temp();
  float t1 = t+3;
  Ultrasonido();

  int i,j,k,m;

  while(distancia>30){
    Ultrasonido();
    for(i=0;i<45;i++)
    {
      servo1.write(i);
      delay(8);
    }
    Ultrasonido();
    if(distancia<30){
      break;
    }
    delay(8);

    for(i=45;i<90;i++)
    {
      servo1.write(i);
      delay(8);
    }
    Ultrasonido();
    if(distancia<30){
      break;
    }
    delay(400);

    for(i=90;i>45;i--)
    {
      servo1.write(i);
      delay(8);
    }
    Ultrasonido();
    if(distancia<30){
      break;
    }
    delay(8);
    for(i=45;i>0;i--)
    {
      servo1.write(i);
      delay(8);
    }
  }
}

```

Figure 8. Programa en Arduino parte 3

```

if (t1>alarma) // si supera el valor definido como alarma
{
  display.setTextSize(1);
  display.setCursor(0,0);
  display.print("Temperatura: ");
  display.setTextSize(2);
  display.setCursor(0,10);
  display.print(t1);
  display.print(" ");
  display.setTextSize(1);
  display.cp437(true);
  display.write(167);
  display.setTextSize(2);
  display.print("C");
  display.setTextSize(2);
  display.setCursor(40,50);
  display.print(" ALERTA ");
  display.display();
  Serial.println("*** Alerta ***");
  Serial.println(" ");

  digitalWrite(rojo, HIGH);
  digitalWrite(verde, LOW);
  delay(3000);
}
else {
  Serial.println(".....");
  digitalWrite(rojo, LOW);
  digitalWrite(verde, HIGH);
  delay(3000);
}
digitalWrite(rojo, LOW);
digitalWrite(verde, LOW);

delay(3000);

```

Figure 10. Programa en Arduino parte 5

## V. SIMULACION EN COPPELIASIM

Para esta simulación, se programó el modelo robótico importado de solidwork con formato URDF, para que cuando una persona se acerque lo suficiente active su programación secuencial determinada

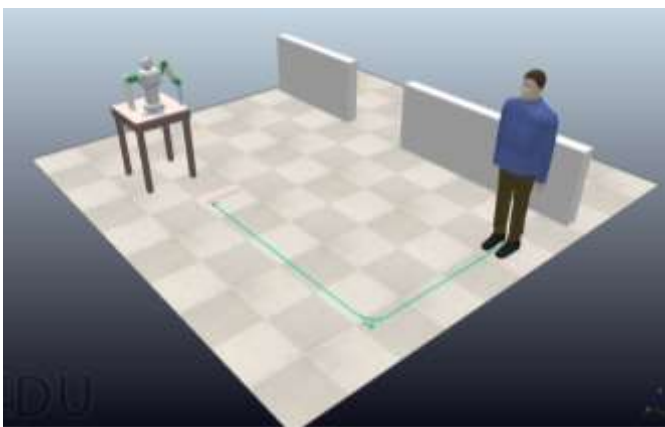


Fig 14. Modelo de simulación en Coppeliassim



Fig 15. Simulación en Reproducción

Luego, la persona levanta su mano para que el robot pueda leer su temperatura, así sabrá si su temperatura es estable o es alta presentando signos de la enfermedad.



Fig 16. Robot midiendo la temperatura

Finalmente, el robot le rociará con alcohol farmacéutico por medio de un mecanismo que se activa por un relay.



Fig 17. Robot en modo de rociamiento

## V. PRUEBAS Y RESULTADOS

Los siguientes resultados se han aplicado a todos los casos posibles que se puedan presentar, tanto la rotación de un servomotor, medición de temperatura por el sensor, visualización de la temperatura y estado de la persona; y finalmente su accionamiento del rociador representado por un LED.

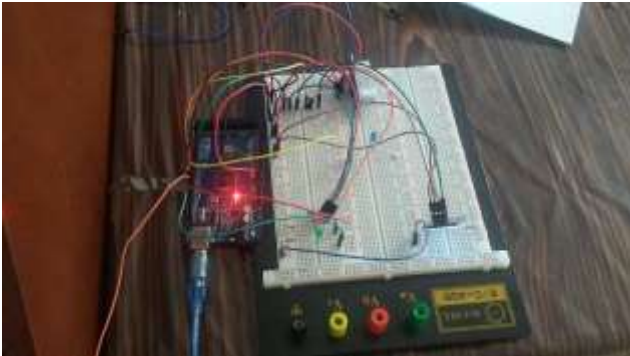


Figure 16. Modelo Físico del proyecto



Figure 17. Circuito alimentador del Servomotor



Figure 18. Sensor de temperatura infrarrojo MLX90615



Figure 19. Temperatura Normal mostrada en el OLED

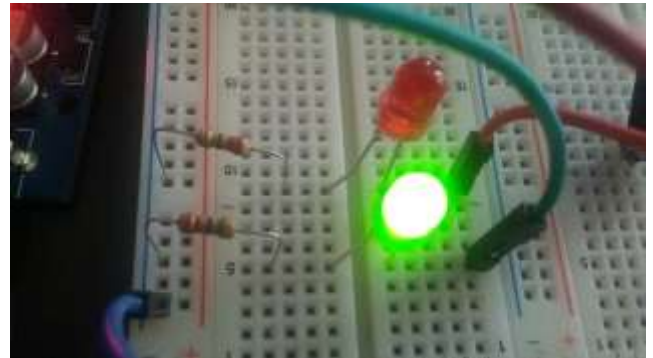


Figure 20. LED indicador de temperatura normal



Figure 21. Temperatura Alta mostrada en el OLED

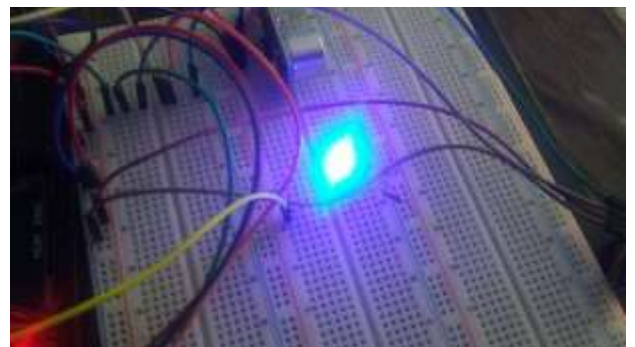
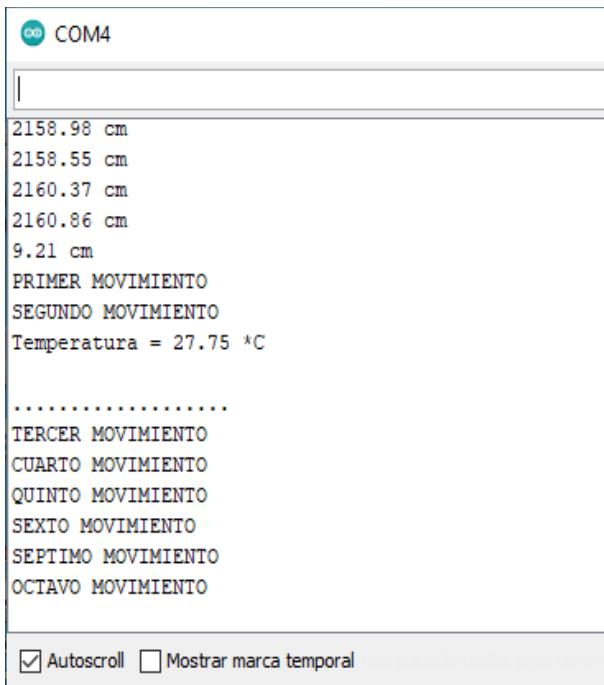


Figure 22. LED que muestra la activación del rociador

The image shows a screenshot of the Arduino Serial Monitor interface. At the top left, there is a port selection dropdown menu showing 'COM4'. Below the port selection is a text input field. The main area of the monitor displays the following text: '2158.98 cm', '2158.55 cm', '2160.37 cm', '2160.86 cm', '9.21 cm', 'PRIMER MOVIMIENTO', 'SEGUNDO MOVIMIENTO', 'Temperatura = 27.75 °C', a line of dots '.....', 'TERCER MOVIMIENTO', 'CUARTO MOVIMIENTO', 'QUINTO MOVIMIENTO', 'SEXTO MOVIMIENTO', 'SEPTIMO MOVIMIENTO', and 'OCTAVO MOVIMIENTO'. At the bottom of the window, there are two checkboxes: 'Autoscroll' which is checked, and 'Mostrar marca temporal' which is unchecked.

```
COM4
|
2158.98 cm
2158.55 cm
2160.37 cm
2160.86 cm
9.21 cm
PRIMER MOVIMIENTO
SEGUNDO MOVIMIENTO
Temperatura = 27.75 °C
.....
TERCER MOVIMIENTO
CUARTO MOVIMIENTO
QUINTO MOVIMIENTO
SEXTO MOVIMIENTO
SEPTIMO MOVIMIENTO
OCTAVO MOVIMIENTO
 Autoscroll  Mostrar marca temporal
```

Figure 23. Visualización del programa en el MonitorSerie de Arduino

## VI. CONCLUSIONES

Al finalizar el proyecto, hemos aprendido que múltiples dispositivos y accesorios pueden ser utilizados para una función determinada en el área Robótica como el accionamiento de servomotores y su ejecución programada al momento de tener lecturas externas por los sensores.

A pesar de que el programa funciona, se puede mejorarlo aún más con líneas de código de control adicionales para detectar si la persona posiciono su mano correctamente o que si la persona se retiró antes de la medición o rociamiento. Además. Se puede mejorar sus movimientos añadiendo nuevos tipos de programación como el software de MatLab para aplicar ecuaciones cinemáticas y dinámicas y se obtenga un manejo de movimientos más precisos y efectivos.

Se tuvo problemas con algunos componentes como el Display LCD16x2, ya que no mostraba ninguna lectura y errores en las librerías descargadas para su aplicación, por lo que en el último momento se decidió cambiar a un Display OLED 0.96.