



Escuela de Educación Técnica N°7
Taller Regional Quilmes
Prácticas Profesionalizantes: Especialidad
Aviónica



CISAR

Escuela: E.E.S.T N°7
T.R.Q.
Curso: 7mo 2da Av



**Escuela de Educación Técnica N°7
Taller Regional Quilmes
Prácticas Profesionalizantes: Especialidad Aviónica**

Integrantes:

- GONZALEZ, Pedro -- D.N.I 44.419.783. -- 7mo 2da Aviónica
- OSORES, Martin -- D.N.I. 44.958.945 -- 7mo 2da Aviónica
- RUOCCO, Julian -- D.N.I. 44.338.599 -- 7mo 2da Aviónica
- PALACIOS, Laureano -- D.N.I. 44.593.048 -- 7mo 2da Aviónica

Fotos del grupo:

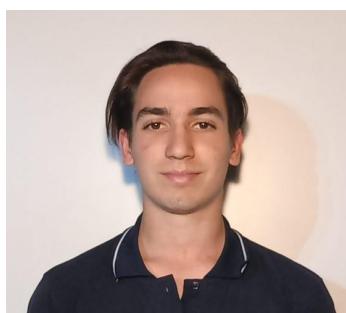


de



GONZALEZ, Pedro

OSORES, Martin



RUOCCO, Julian



PALACIOS, Laureano

Tutores:

- MEDINA,
- BIANCO
- SBRUZZI
- CARLASSARA



Inicio de actividades:

- 8 de marzo 2021
- 37 semanas trabajadas
- 300 horas de trabajo total
- Tareas abarcadas:
 - Investigación de soluciones
 - Investigación de componentes
 - Diseño de la estructura
 - Control de Ingreso de personas:
 - Sistema de Actividad
 - Difusión y establecimiento en redes sociales
 - Creacion de pagina web



Índice:

Objetivo:	5
Temática:	5
Descripción general:	5
Alcance:	6
Segmento de Usuario:	7
Información técnica:	7
Detección de ingresantes:	7
Chequeo de usuarios:	9
Asistente Virtual:	10
Base de Datos:	13
Medición de Temperatura:	14
Medición de Oxígeno en Sangre:	17
Descripción:	17
Utilizamos el MAX 30102 que incorpora las funciones de pulsímetro y oxímetro en un único integrado que podemos usar junto con un procesador como Arduino.	17
Esterilización de manos:	22
Sanitización de Ozono:	24
Alfombra virucida:	27
Desinfección de Objetos:	28
Especificaciones Técnicas:	28
Bibliografía:	29



Objetivo:

Nuestro proyecto “**Cisar**” consiste en un sistema de desinfección y monitoreo eficaz . Está pensado principalmente para que los lugares de gran cantidad de personas, que puedan tener un sistema de desinfección eficaz, controlada y organizada, previniendo el contagio de Covid-19, de todo el volumen de personas de manera automatizada, a través del uso adecuado del barbijo, control de temperatura corporal, la oxigenación en sangre e incitación a esterilizar las manos, el cuerpo en su totalidad y los objetos personales.

Para esto se construirá una cabina sanitizante y un control esterilizador de objetos personales.

Temática:

Debido a la situación que estamos viviendo actualmente a nivel mundial, se debe tener una gran responsabilidad en relación a los cuidados que tomamos y más aún en lugares donde concurre un gran volumen de personas. Ante esta problemática nos topamos con un modelo de cuidado ineficiente, por ello nuestra propuesta consta de un aparato que nos ayude a lograr esos cuidados de la manera más eficiente posible, reduciendo considerablemente las posibilidades de contagio, logrando un eficiente control de multitudes. Este proyecto está pensado puntualmente para colegios y empresas donde tienen gran volumen de alumnos, hipermercados y fábricas.

Descripción general:

Nuestra propuesta ante el Coronavirus y su repercusión a nivel mundial, es una cabina con la capacidad de tomar la temperatura de la persona que ingresa, sensar la oxigenación en sangre, desinfectar con un rociado alcohol



en las manos, y de detectar la correcta posición del barbijo mediante una cámara, con Inteligencia Artificial .

Para su funcionamiento en régimen normal, el sistema estará enchufado a un cable de red común, de 220v.

Debe tener una cámara que mediante una Raspberry, y un programa ya entrenado de Inteligencia Artificial; identificará la correcta posición del barbijo, en este caso, le mostrará a la persona en una pantalla si está todo dentro de los parámetros correspondientes (Temperatura corporal, Oxigenación en sangre y el tapa boca), dándole acceso a la institución. De lo contrario, le indicará cómo debe ir colocado el tapabocas o cómo proceder ante la detección de una temperatura considerada como fiebre y dará aviso a las autoridades de la institución para que tengan la precaución correspondiente.

Se llevará un registro mediante el cual quedarán asentadas la cantidad de personas que ingresen a la institución, con su horario de ingreso respectivo, su temperatura, y si tenía bien colocado el tapabocas en el primer escaneo que realiza el sistema.

Alcance:

Desarrollar y construir una cabina sanitizante para el uso de personal en Aeropuertos, Talleres aeronáuticos, escuelas, fábricas, aeródromos, hospitales, hoteles, etc. Que será estructuralmente de armado modular y efectué la aclimatación de la persona que ingrese a una temperatura determinada, desinfecte con una corriente de aire las vestimentas y con alfombras los calzados, también identifique con reconocimiento facial la correcta posición de los tapabocas y dosifique alcohol en gel para las manos de la persona que ingresa, realizando todo eso en forma automática.



Segmento de Usuario:

Ventajas

- Desinfecta a los usuarios en cuestión de segundos
- No utiliza químicos nocivos para la salud
- Promueve la higiene personal e incitación a lavarse las manos
- Fomenta el cuidado del medio ambiente con el uso de materiales reciclados
- Garantiza espacios completamente libres de virus

Información técnica:

→ Detección de ingresantes:

I. Descripción:

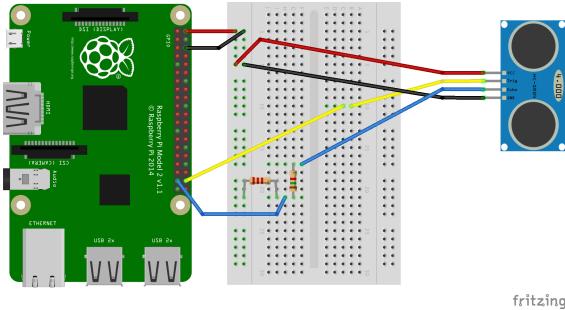
El sensor ultrasónico realiza mediciones mediante el uso de ondas ultrasónicas. Su funcionamiento es simple: cuando el impulso de onda ultrasónica encuentra un objeto, es reflejado y así es medido en distancia = velocidad * tiempo.

II. Funcionamiento:

La tecnología ultrasónica funciona basándose en principios acústicos. El sensor emite pulsaciones acústicas cortas a través del aire, y de esta forma detecta objetos que después reflejará. Lo que mide el sensor es el tiempo entre emitir el pulso y la recepción de este.

Esta es la razón por la que el sensor ultrasónico es de los instrumentos más confiables en medición. Casi todo tipo de material es capaz de reflejar ondas acústicas o sonido, haciendo que las mediciones sean limpias.

III. Esquema Eléctrico:



IV. Código:

```
import RPi.GPIO as GPIO
import time

TRIG = 38
ECHO = 40
GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
try:
    while True:

        GPIO.output(TRIG, GPIO.LOW)
        time.sleep(0.5)

        GPIO.output(TRIG, GPIO.HIGH)
        time.sleep(0.00001)
        GPIO.output(TRIG, GPIO.LOW)

    while True:
        pulso_inicio = time.time()
        if GPIO.input(ECHO) == GPIO.HIGH:
            break

    while True:
        pulso_fin = time.time()
        if GPIO.input(ECHO) == GPIO.LOW:
            break

    # Tiempo medido en segundos
    duracion = pulso_fin - pulso_inicio

    distancia = (34300 * duracion) / 2
    print("Distancia: %.2f cm" % distancia)
```



```
if distancia < 50:  
    print ("Se ha detectado un objeto cerca")  
  
elif distancia > 50:  
    print ("No hay objeto en los proximos 50 cm")  
  
finally:  
    # Reiniciamos todos los canales de GPIO.  
    GPIO.cleanup()
```

→ Chequeo de usuarios:

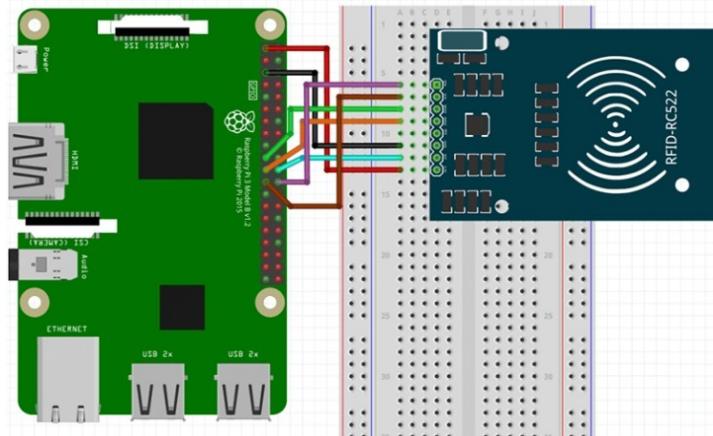
I. Descripción:

RFID o identificación por radiofrecuencia es un sistema de almacenamiento y recuperación de datos remotos que usa dispositivos denominados etiquetas, tarjetas o transpondedores RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto mediante ondas de radio.

II. Funcionamiento:

La antena en un tag pasivo funciona captando la energía de las ondas radioeléctricas emitidas por el interrogador en una determinada frecuencia para la que es dimensionada, de tal forma que resuena esa frecuencia y capta las ondas de radio básicamente en esa frecuencia. Dependiendo de la frecuencia a la que estemos trabajando la antena será físicamente distinta, debido a que ha de acoplarse a determinadas longitudes de onda. Así, en LF la antena suele estar compuesta por bobinas con un elevado número de espiras realizadas con hilo esmaltado bobinado en anillos o bien sobre núcleos de ferrota, lo que permite unas menores dimensiones físicas de la antena, siendo el acoplamiento básicamente inductivo.

III. Esquema Eléctrico:



IV. Código:

```
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522

reader = SimpleMFRC522()

try:
    id, text = reader.read()
    print(id)

    print(type(text))
finally:
    GPIO.cleanup()
```

→ Asistente Virtual:

I. Descripción:

La interfaz del asistente es puramente textual, y lo controlarás enviándole mensajes que contengan determinados comandos o instrucciones que el entienda para poder reaccionar a ellos.

Este bot está totalmente automatizado. Nosotros no estamos detrás, sino que está preparado para dar determinados tipos de respuestas dependiendo del comando que les escriba usted. Al ser un bot conversacional, incluye botones en sus



respuestas con los que enviar comandos o peticiones sin necesidad de escribirlas.

II. Funcionamiento:

En primer lugar, tendrás que buscarlos como si fueran otro tipo de usuario, utilizando @Cisar_Bot o mediante nuestro código QR.



Cuando entres en una conversación con el bot, oprimiendo el botón */start* te saludará y solicitará que presiones */ayuda* para ver cómo funciona, mostrándote la lista de comandos.

El bot cumple con las siguientes funciones principales:

- Registrarse como alumno (pulsar /registro)
- Chequeo del sistema (pulsar /ingresar)
- Ocuparse del posible sospechoso (pulsar /voy)

chequeo manual, del registro. Debajo de la descripción tendrás que pulsar algún botón. el que pone *Start*, y que sirve para que envíes automáticamente el comando de inicio del Bot. Al hacerlo, el bot te responderá con un mensaje explicando de

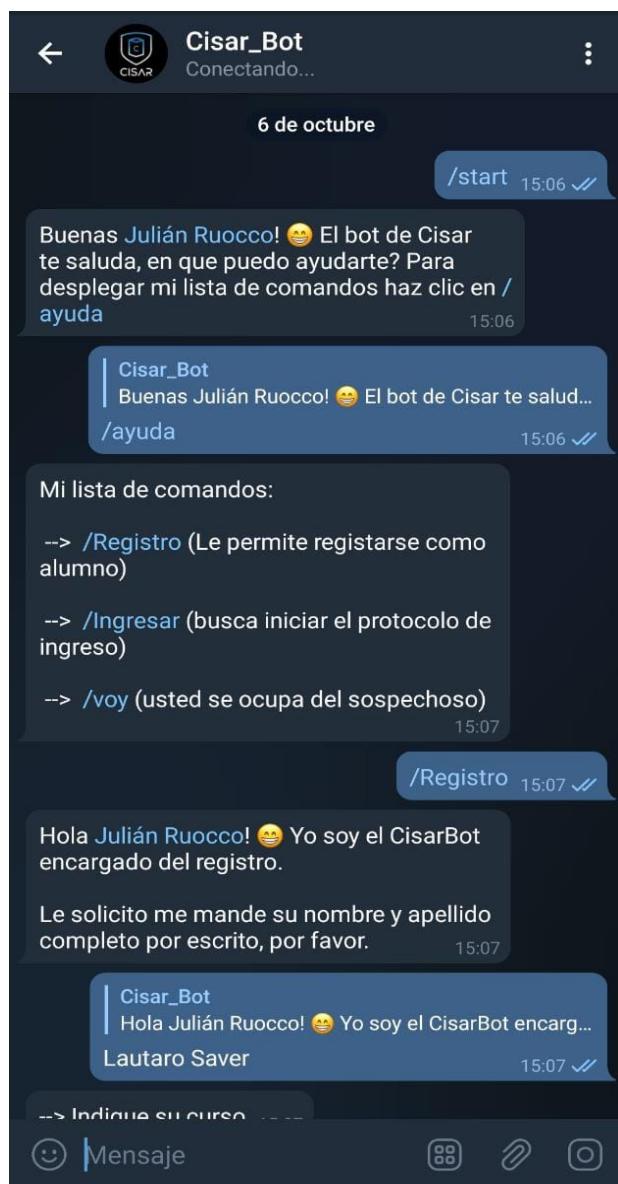


**Escuela de Educación Técnica N°7
Taller Regional Quilmes
Prácticas Profesionalizantes: Especialidad Aviónica**

nuevo su funcionamiento y diciéndote los principales comandos para empezar a utilizarlo.

Y luego ya está. Empezarás a utilizar sus comandos de funcionamiento, y puede que en sus respuestas tengas otros comandos para ir navegando por su menú de contenidos y sus mensajes.

III. Software:





→ Base de Datos:

I. Descripción:

Una base de datos es una herramienta para recopilar y organizar información. Las bases de datos pueden almacenar información sobre personas, productos, pedidos u otras cosas. Muchas bases de datos comienzan como una lista en una hoja de cálculo o en un programa de procesamiento de texto.

II. Funcion:

Utilizamos sqlite3 para crear una base de datos y cargar la información de los usuarios de la cabina, de esta manera, al momento que un ingresante apoya la tarjeta sobre el sensor Rfid, podemos corroborar si está en el sistema para darle el ingreso a la cabina o rechazarlo.

III. Código:

Crear base de datos:

```
def crearBaseDeDatos():
    sqliteConnection = sqlite3.connect("home/pi/DesktopPrincipal/BASE_DE_DATOS_CISAR.db")

    cursor = sqliteConnection.cursor()
    cursor.execute("""CREATE TABLE usuarios(
        nombre text,
        curso integer,
        division integer,
        especialidad integer,
        dni integer,
        numero_tarjeta_rfid integer)""");
```

Subir base de datos:

```
def subirBaseDeDatos (update, context) :
```



```
sqliteConnection=sqlite3.connect("/home/  
pi/Desktop/Principal/BASE_DE_DATOS_CISAR.db')  
cursor = sqliteConnection.cursor()  
cursor.execute ("INSERT INTO usuarios VALUES  
(?,?,?,?,?,?)", (nombre_apellido, curso,  
division, especialidad, dni,  
numero_tarjeta rfid))  
  
emoii = "\N{flexed biceps}"  
update.message.reply_text(  
f"Datoscargados satisfactoriamente! {emoji} ")  
  
sqliteConnection.commit()  
sqliteConnection.close()
```

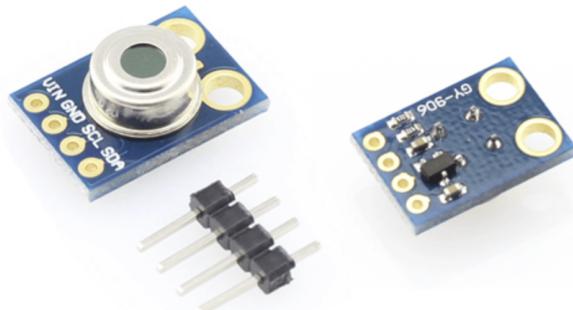
→ Medición de Temperatura:

I. Descripción:

Nosotros utilizamos el MLX90614 que es un sensor de temperatura infrarroja sin contacto. Es posible conectar estos sensores con un autómata o procesador como Arduino para medir la temperatura de un objeto a distancia.

La comunicación se realiza a través de SMBus, un subconjunto de bus I2C, por lo que resulta sencilla su lectura, y es posible conectar más de un sensor de forma simultáneamente.

Este tipo de termómetros infrarrojos tienen un gran número de aplicaciones, incluyendo sistemas de control de temperatura en instalaciones térmicas en edificios, control industrial de temperatura, detección de movimiento, y aplicaciones de salud.



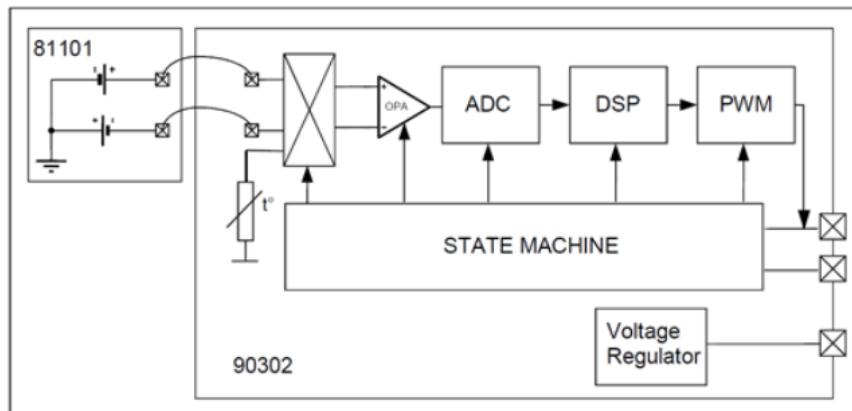


II. Funcionamiento:

El MLX90614, toma la radiación que emiten los objetos, cuyo espectro es proporcional a la temperatura, entonces toma esta radiación y su salida es una señal eléctrica proporcional a la temperatura de todos los objetos en su campo de visión.

Internamente el MLX90614 está constituido con un chip de silicio con una fina membrana micro mecanizada sensible a la radiación infrarroja, junto con la electrónica necesaria para amplificar y digitalizar la señal y calcular la temperatura.

El conjunto incluye un amplificador de bajo ruido, un conversor ADC de 17 bits, un DSP (procesador digital de señal) y compensación de temperatura ambiente.



El MLX90614 viene calibrado de fábrica en un amplio rango de temperaturas: -40 a 85 °C para la temperatura ambiente y -70 a 382 °C para la temperatura de objetos. La precisión estándar es de 0.5 °C referente a la temperatura ambiente, aunque existen versiones médicas que ofrecen una resolución de 0.1°C en temperaturas entre 35-38°C.

El MLX90614 dispone de dos modos de salida. El estándar es SMBus, un conjunto del I2C, con una resolución de 0.02°C. También puede emplear una salida PWM de 10 bits para mediciones continuas, aunque con menor resolución 0.14°C.

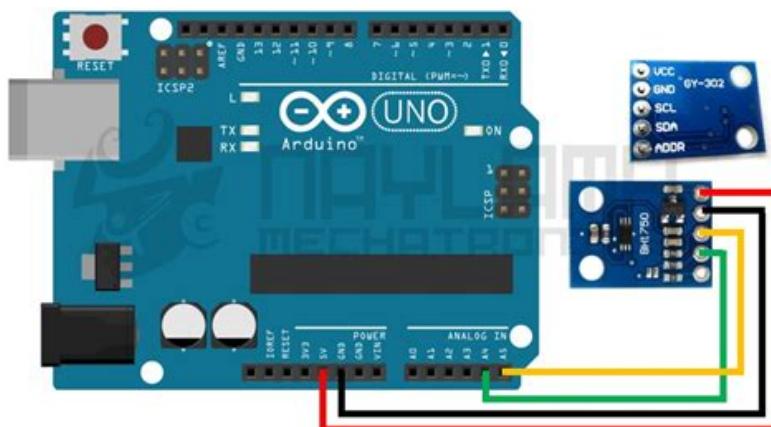
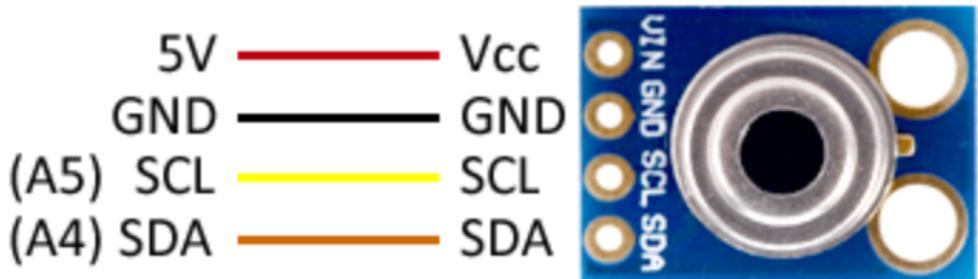
Es importante tener en cuenta que la lectura del sensor solo es estable cuando el sensor se encuentra en equilibrio



térmico con el ambiente. También puede afectar la suciedad en la ventana del sensor.

III. Esquema Eléctrico:

La conexión es sencilla, simplemente alimentamos el módulo desde Arduino mediante GND y 5V y conectamos el pin SDA y SCL de Arduino con los pines correspondientes del sensor.



IV. Código:

Temperatura_9.00 §

```
#include <Wire.h>
#include <Adafruit_MLX90614.h>
//#include <i2cmaster.h>

Adafruit_MLX90614 mlx = Adafruit_MLX90614();
int flagtemp = 8;
int flagtemp_dos = 5;
byte MLXAddr = 0x5A<<1;

const int max = 37.5;

void setup() {
  Serial.begin(9600);
  mlx.begin();
  Serial.println("Adafruit MLX90614 test");
  pinMode(flagtemp, OUTPUT);
  pinMode(flagtemp_dos, OUTPUT);
}

void loop() {
```



Temperatura_9.00 \$

```
float medicion = mlx.readObjectTempC();

float medicion_corregida = medicion * 1.152;

//Serial.print("Ambient = "); Serial.print(mlx.readAmbientTempC());
//Serial.print("C\tObject = ");
Serial.print(mlx.readObjectTempC()); Serial.println("C");
Serial.print(medicion_corregida); Serial.println("C");
//Serial.print("F\tObject = "); Serial.print(mlx.readObjectTempF()); Serial.println("*F");
Serial.println();

if (medicion_corregida > max){
    digitalWrite(flagtemp, HIGH);
    Serial.println("No puede ingresar");
    delay (60000);

}
else if (medicion_corregida < max) {
    digitalWrite(flagtemp, LOW);
    Serial.println("Podes ingresar mostro");
    digitalWrite(flagtemp_dos, HIGH);
    delay(200);
    digitalWrite(flagtemp_dos, LOW);
}
delay(1000);
}
```

→ Medición de Oxígeno en Sangre:

I. Descripción:

Utilizamos el MAX 30102 que incorpora las funciones de pulsímetro y oxímetro en un único integrado que podemos usar junto con un procesador como Arduino.

La serie MAX 30102 es un sensor óptico, que basa su funcionamiento en el distinto comportamiento que la sangre tiene ante la luz, en función de su grado de saturación de oxígeno.

Para ello, el MAX 30102 incorpora dos LED, uno de espectro rojo y otro de infrarrojo. El MAX 30102 se pone sobre la piel, por ejemplo en el dedo o la muñeca. El sensor detecta la luz reflejada, y determina el grado de saturación.

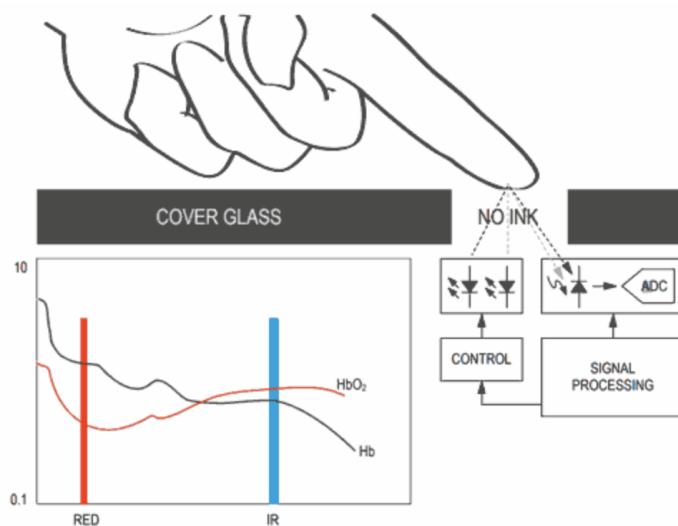
La comunicación con el MAX 30102 se realiza a través de bus I2C, por lo que es muy sencillo conectarlo a un procesador como Arduino.

El MAX 30102 requiere una doble alimentación de 1.8V para la lógica, y 3V3 para los leds. Normalmente lo encontraremos en módulos de 5V que incorporan la adaptación de nivel necesaria.



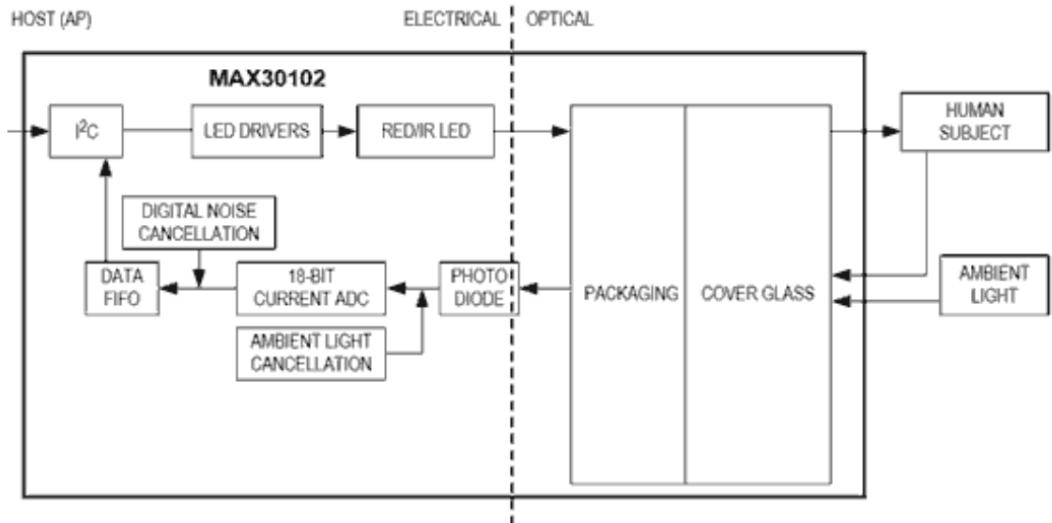
II. Funcionamiento:

La pulsioximetría óptica es un método no invasivo para determinar el porcentaje de saturación de oxígeno en sangre. Su funcionamiento se basa en que la hemoglobina (Hb) y la hemoglobina saturada (oxihemoglobina, HbO₂) tienen distintos coeficientes de absorción de luz para distintas longitudes de onda.



La sangre oxigenada absorbe mayor cantidad de luz infrarroja, mientras que la sangre poco oxigenada absorbe mayor luz roja. Además, en las partes del cuerpo donde la piel es suficientemente fina y por debajo pasan los vasos sanguíneos, es posible emplear esta diferencia para determinar el grado de saturación.

El MAX 30102 incorpora: dos LED, uno de espectro rojo y otro de infrarrojo, un ADC de 18 bits y una frecuencia de muestreo de 50 sps (samples per second) a 3200spf.



También dispone de la electrónica necesaria para la amplificación y filtrado de la señal, cancelación de luz ambiental y rechazo a frecuencias de 50-60Hz (luz artificial) y compensación de temperatura.

El consumo del módulo es de hasta 50 mA durante la medición, aunque la intensidad puede ajustarse en la programación, con un modo de bajo consumo de 0.7 μ A durante las mediciones.

III. Esquema Eléctrico:

La conexión es sencilla, simplemente alimentamos el módulo desde Arduino mediante GND y 5V y conectamos el pin SDA y SCL de Arduino con los pines correspondientes del MAX 30102.

5V	—	Vcc
GND	—	GND
A5	—	SCL
A4	—	SDA





IV. Código:

```
#include <Wire.h>
#include "MAX30105.h"
#include "spo2_algorithm.h"

MAX30105 particleSensor;
#define MAX_BRIGHTNESS 255
int32_t bufferLength;
int32_t spo2;
int8_t validSPO2;
int8_t validHeartRate;

byte pulseLED = 11;
byte readLED = 13;

void setup()
{
    Serial.begin(115200);
    pinMode(pulseLED, OUTPUT);
    pinMode(readLED, OUTPUT);
    if (!particleSensor.begin(Wire, I2C_SPEED_FAST))
    {
        Serial.println(F("MAX30105 was not found. Please check
wiring/power."));
        while (1);
    }

    Serial.println(F("Attach sensor to finger with rubber band. Press
any key to start conversion"));
    while (Serial.available() == 0) ;
    Serial.read();
    byte ledBrightness = 60;
    byte sampleAverage = 4;
    byte ledMode = 2;
    byte sampleRate = 100;
    int pulseWidth = 411;
    int adcRange = 4096;
```



```
particleSensor.setup(ledBrightness, sampleAverage, ledMode,
sampleRate, pulseWidth, adcRange); //Configure sensor with
these settings
}

void loop()
{
    bufferLength = 100;
    while (1)
    {
        for (byte i = 25; i < 100; i++)
        {
            redBuffer[i - 25] = redBuffer[i];
            irBuffer[i - 25] = irBuffer[i];
        }
        for (byte i = 75; i < 100; i++)
        {
            while (particleSensor.available() == false)
                particleSensor.check();

            digitalWrite(readLED, !digitalRead(readLED));

            redBuffer[i] = particleSensor.getRed();
            irBuffer[i] = particleSensor.getIR();
            particleSensor.nextSample();

            Serial.print(F("red="));
            Serial.print(redBuffer[i], DEC);
            Serial.print(F(", ir="));
            Serial.print(irBuffer[i], DEC);

            Serial.print(F(", HR="));
            Serial.print(heartRate, DEC);

            Serial.print(F(", HRvalid="));
            Serial.print(validHeartRate, DEC);

            Serial.print(F(", SPO2="));
            Serial.print(spo2, DEC);
```



```
Serial.print(F(", SPO2Valid="));
Serial.println(validSPO2, DEC);
}

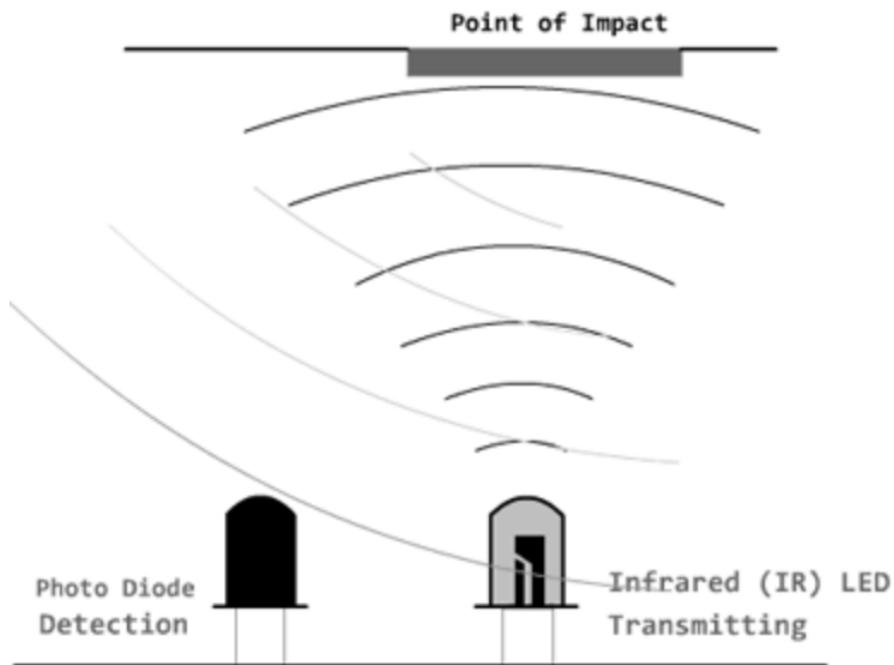
maxim_heart_rate_and_oxygen_saturation(irBuffer,
bufferLength, redBuffer, &spo2, &validSPO2, &heartRate,
&validHeartRate);
}
}
```

→ Esterilización de manos:

I. Descripción:

Nosotros utilizamos un detector de obstáculos con un sensor infrarrojo arduino. Donde su principal función es detectar la presencia de un objeto mediante la reflexión que produce en la luz. El uso de luz infrarroja (IR) es simplemente para que esta no sea visible para los humanos.

Constitutivamente son sensores sencillos. Se dispone de un LED emisor de luz infrarroja y de un fotodiodo (tipo BPV10NF o similar) que recibe la luz reflejada por un posible obstáculo.



II. Funcionamiento:

Los detectores de obstáculo suelen proporcionarse con una placa de medición estándar con el comparador LM393, que



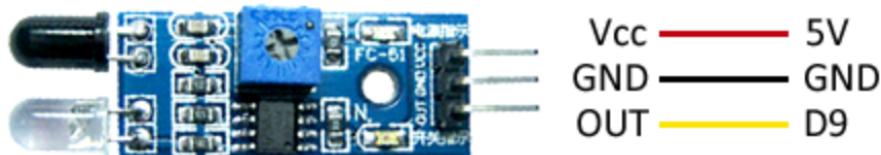
permite obtener la lectura como un valor digital cuando se supera un cierto umbral, que se regula a través de un potenciómetro ubicado en la placa.

Este tipo de sensores actúan a distancias cortas, típicamente de 5 a 20mm. Además la cantidad de luz infrarroja recibida depende del color, material, forma y posición del obstáculo, por lo que no disponen de una precisión suficientemente para proporcionar una estimación de la distancia al obstáculo.

Pese a esta limitación su principal función es la detección de las manos, para así accionar el gatillo del atomizador de alcohol

III. Esquema Eléctrico:

El montaje es sencillo. Alimentamos el módulo a través de Vcc y GND conectándose, respectivamente, a la salida de 5V y GND en Arduino.



IV. Código:

```
const int sensorPin = 9;

void setup() {
    Serial.begin(9600); //iniciar puerto serie
    pinMode(sensorPin , INPUT); //definir pin como entrada
}

void loop(){
    int value = 0;
    value = digitalRead(sensorPin ); //lectura digital de pin

    if (value == HIGH) {
        Serial.println("Detectado obstaculo");
    }
    delay(1000);
}
```



→ Sanitización de Ozono:

I. Descripción:

Nosotros usamos el MQ131, que es un dispositivo diseñado para detectar la cantidad de ozono. Este sensor está proporcionado en una placa de medición estándar con el comparador LM662 o similar, que permite obtener la lectura tanto como un valor analógico, como un valor digital cuando se supera un cierto umbral regulado a través de un potenciómetro ubicado en la placa.

II. Funcion:

El sensor MQ 131, está compuesto por un sensor electroquímico, que varía su resistencia al estar en contacto con el Ozono.

La familia de los sensores MQ son dispositivos con alta inercia, es decir, la respuesta necesita tiempos largos para estabilizarse tras un cambio de concentración de los gases medidos. Ello es debido a la necesidad física de que el gas abandone el material sensible, lo cual es un proceso lento.

El MQ131 dispone de un calentador necesario para elevar la temperatura del sensor, y que sus materiales adquieran la sensibilidad. Mientras el calentador no alcance la temperatura de funcionamiento, la lectura del sensor no será fiable.

Para que de una respuesta, $\leq 1S$, el calentamiento tiene que ser entre los 30-50 min, y el tiempo de calentamiento teórico es de 48hs. Este tiempo de calentamiento, genera que su consumo sea elevado, llegando hasta los 900 mW. Esto es superior a la potencia que puede suministrar el regulador de Arduino, por lo que será necesario proporcionar una fuente de alimentación externa

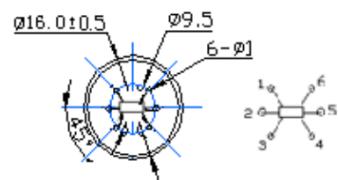
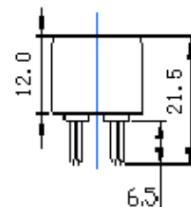
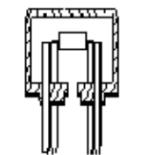
MQ-131

Ozono

6V

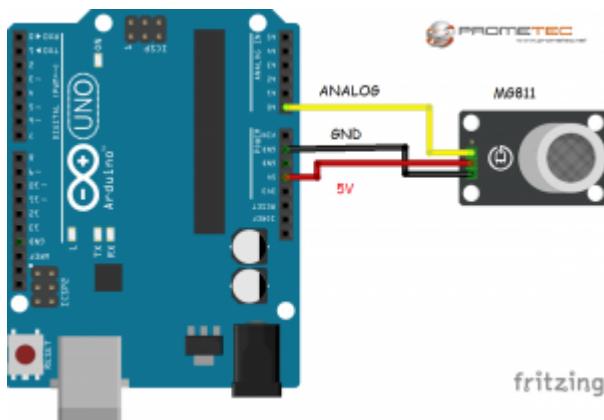


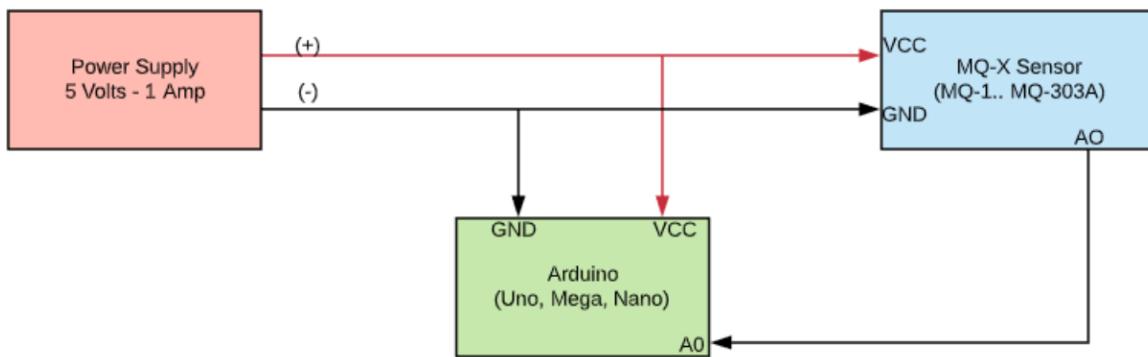
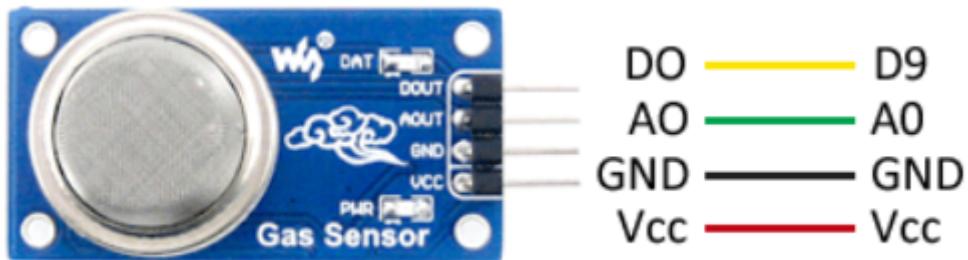
Model		MQ131	
Sensor Type		Semiconductor	
Standard Encapsulation		Plastic cap	
Target Gas		Ozone	
Detection range		10~1000ppb Ozone	
Standard Circuit Conditions	Loop Voltage	V_c	$\leq 24V$ DC
	Heater Voltage	V_H	$5.0V \pm 0.1V$ AC or DC
	Load Resistance	R_L	Adjustable
Sensor character under standard test conditions	Heater Resistance	R_H	$34\Omega \pm 3\Omega$ (room tem.)
	Heater consumption	P_H	$\leq 900mW$
	Sensitivity	S	$R_s(\text{in } 200\text{ppb O}_3)/R_s(\text{in air}) \geq 2$
	Output Voltage	ΔV_s	$\geq 1.0V (\text{in } 200\text{ppb O}_3)$
	Concentration Slope	α	$\leq 0.6 (R_{500\text{ppb}}/R_{100\text{ppb O}_3})$
Standard test conditions	Tem. Humidity		$20^{\circ}\text{C} \pm 2^{\circ}\text{C}; 55\% \pm 5\% \text{RH}$
	Standard test circuit		$V_c: 5.0V \pm 0.1V; V_H: 5.0V \pm 0.1V$
	Preheat time		Over 48 hours



III. Esquema Eléctrico:

El esquema eléctrico es sencillo. Alimentamos el módulo conectando GND y 5V a los pines correspondientes de Arduino. En el caso del MQ131, para emplear el valor analógico, simplemente conectaremos la salida AO (AOUT) del sensor a una entrada analógica de Arduino.





IV. Código:

```
#include <MQ131.h>
void setup() {
    Serial.begin(115200);
    MQ131.begin(2,A0, LOW_CONCENTRATION, 1000000);

    Serial.println("Calibration parameters");
    Serial.print("R0 = ");
    Serial.print(MQ131.getR0());
    Serial.println(" Ohms");
    Serial.print("Time to heat = ");
    Serial.print(MQ131.getTimeToRead());
    Serial.println(" s");
}

void loop() {
    Serial.println("Sampling...");
    MQ131.sample();
    Serial.print("Concentration O3 : ");
    Serial.print(MQ131.getO3(PPM));
    Serial.println(" ppm");
    Serial.print("Concentration O3 : ");
    Serial.print(MQ131.getO3(PPB));
```



```
Serial.println(" ppb");
Serial.print("Concentration O3 : ");
Serial.print(MQ131.getO3(MG_M3));
Serial.println(" mg/m3");
Serial.print("Concentration O3 : ");
Serial.print(MQ131.getO3(UG_M3));
Serial.println(" ug/m3");

delay(1000);
}
```

→ Alfombra virucida:

I. Descripción:

Usamos una alfombra virucida porque en los suelos se acumulan una enorme cantidad de bacterias, debido al contacto directo con la suela de los zapatos o por contaminación ambiental, por lo que es más que recomendable el uso de una alfombra de desinfección o alfombra antibacteriana con pulverización de líquido desinfectante hidroalcohólico o en detergente de monodosis.

II. Funcionamiento:

La alfombra textil con base de caucho de nitrilo debe ser la primera barrera desinfectante en la entrada. Nosotros pulverizamos el producto sobre la alfombras de manera periódica, según el tránsito de personas. En esta alfombra, puede contener un descontaminante con líquido hidroalcohólico o en detergente de monodosis.

Utilizaremos una alfombra desinfectante para la entrada, además haciendo una limpieza habitual con agua y detergentes limpiadores, tanto de los felpudos como del suelo.

Líquido desinfectante (Producto aprobado por el Ministerio de Salud de la Nación): Nosotros usamos Bacterium Concentrado B-56 de 1Lt, ya que si lo diluimos nos rinde un total de 60 Lt. Este químico tiene como activo principal amonio cuaternario de 5ta generación y como activo secundario, alcohol



isopropílico. Este producto puede llegar a tener un periodo de desinfección de 4 horas.

Su uso principal es eliminar microorganismos ambientales, actuando como un fungicida, bactericida y virucida. Además se puede aplicar en pisos, paredes, mesadas, picaportes, alfombras sanitizantes, artefactos, textiles, bolsas, envoltorios.

Nosotros lo usamos de la siguiente forma:

Diluimos 17ml en 1lt de agua, aplicándolo con una esponja, trapo o spray a la superficie para desinfectar, dejándolo 2 minutos para que actúe.

→ Desinfección de Objetos:

Al ingresar una persona, primero tiene que dejar sus pertenencias en la máquina sanitizante de objetos, donde tendrá que seguir con el protocolo de ingreso. Para esto utilizaremos una lámpara led UV C y unos espejos que refractan la luz germicida para lograr una cobertura total.

→ Especificaciones Técnicas:

Nuestra cabina cuenta con las siguientes características:

- A nivel estructural:
 - Perfiles tipo C Galvanizados
 - Planchas de PVC 20 cm de ancho
 - Varillas tipo J, donde se colocan las planchas
 - 2 puertas
 - 2 Planchas de PVC, 1m ancho y 1,40m largo, estas se utilizan para evitar que escape la mayor cantidad de ozono
 - Alfombra Virucida
 - Pantalla
 - Rociador
 - Generador de Ozono
 - 2 Coolers
- A nivel Electrónico:
 - 2 Sensores MLX90614 (temperatura)
 - 1 Sensor MAX 30102 (Pulsómetro y Oxímetro)



- 1 Módulo RFID
- 4 Tarjetas y Llaveros RFID
- 1 Módulo Infrarrojo (Sist de Alcohol)
- 1 MQ131 Arduino low (10-1000ppb)
- A nivel de Software:
 - Raspberry Pi 4
 - Arduino Mega
 - Python 3.8
- Apartado de Sanitización de Objetos
 - Luces UV
 - Espejos

→ Bibliografía:

Sensor de Ozono

- [https://trello.com/1/cards/6155baecb7f38c461dd88732/attachments/6155bb2c22c05e014d41b372/download/mq131-\(low-concentration\)-ver1_3-manual.pdf](https://trello.com/1/cards/6155baecb7f38c461dd88732/attachments/6155bb2c22c05e014d41b372/download/mq131-(low-concentration)-ver1_3-manual.pdf)
- https://articulo.mercadolibre.com.ar/MLA-896574020-sensor-de-ozono-mq131-arduino-para-desinfeccion- JM#position=1&search_layout=stack&type=item&tracking_id=244faaa6-4b7a-4a96-bd53-27630bf3fb32