

# SISTEMA DE PROXIMIDAD

Autor, *Gustavo Alexander Ramos Benavides*

Abstract-The development and construction of an electronic device based on electronic microcontrollers is described, in which Python will be implemented to give programmed instructions; This will be able to detect in a given space the movement or approach of a person or object, this proximity reading will be continuous and if the detection is made, an alert method will be activated for the user through notification on the screen and by email. The main component is an ESP32 card that allows the connection of several electronic modules that provide us with a favorable characteristic for the device in question, such as the proximity sensor hc-sr04, which has a detection range: 2cm to 400 cm, an oled 1306 screen and a button, the union of the electronic components with the Python programming using libraries and functions which will give the corresponding instructions for the continuous monitoring of the place, resulting in a device.

Se describe el desarrollo y construcción de un dispositivo electrónico basado en micro controladores electrónicos, en el cual se implementará Python para dar instrucciones programadas; este será capaz de detectar en un espacio determinado el movimiento o acercamiento de una persona u objeto, esta lectura de la proximidad será continua y en caso de hacer la detección se activará un método de alerta para el usuario mediante notificación en pantalla y al correo electrónico. El principal componente es una tarjeta **ESP32** que permite la conexión de varios módulos electrónicos que nos brinden una característica favorable para el dispositivo en mención como es el Sensor de proximidad **hc-sr04**, el cual cuenta con un rango de detección: 2cm a 400 cm, una pantalla **oled 1306** y un pulsador, la unión de los componentes electrónicos con la programación en Python dando uso de librerías y funciones las cuales darán las instrucciones correspondientes para el monitoreo continuo del lugar nos dan como resultado un dispositivo.

## INTRODUCCIÓN

Se busca implementar mediante el uso de dispositivos electrónicos; tarjetas microcontroladores y las herramientas disponibles para la construcción física, parte electrónicas (sensores, cableado, pulsadores, fuentes modulares de energía) usando Python lenguaje de programación que nos servirá para generar las instrucciones de funcionamiento; Python que es un lenguaje de programación de propósito general, que es otra forma de decir que puede ser usado para casi todo.

Lo más importante es que se trata de un lenguaje interpretado, lo que significa que el código escrito no se traduce realmente a un formato legible por el ordenador en tiempo de ejecución y MicroPython se ejecuta en varios microcontroladores y no hay grandes limitaciones para trasladar MicroPython a más de microcontroladores siempre que haya suficiente memoria RAM, Flash y potencia de procesamiento para ejecutar el intérprete, usando apis-rest que nos van permitir la comunicación o salida de información que genere el dispositivo por medio de un servicio SMTP es quizás el protocolo de correo más conocido. Sus siglas significan Simple

Mail Transfer Protocol, en español significa Protocolo Simple de Transporte de Correo. Es un protocolo usado para enviar correo desde un servidor a otro destino para dar notificación de estados y alertas sobre el un monitoreo del ambiente.

## I. PROBLEMÁTICA

En la actualidad en Colombia las alertas en los hogares o locales comerciales respecto a la seguridad es muy alta, la detección de movimiento dentro del lugar un problema de bajo alcance ya que muchas veces tienen instalado un sistema de monitoreo local, el cual falla en fallos de energía, falta de mantenimientos preventivos, además de los errores humanos.

El no contar con un sistema de detección de movimiento para dar alerta de posibles intrusos o incluso roedores, llevan en sí riesgos, la gran mayoría de hurtos los realizan quitando la energía del lugar y dejando sin seguridad.

Quienes tienen mayor riesgo son pequeños negocios, los cuales por falta de capital no se logra implementar un sistema de seguridad y queda a la deriva de los ladrones.

## II. OBJETIVO GENERAL

Diseñar e implementar un sistema de seguridad que detecte movimiento para locales comerciales y hogares a través de lenguaje de alto nivel python y plataformas de desarrollo basados en microcontroladores para disminuir el hurto o saqueos en horarios nocturnos, el cual realice un monitoreo constante del espacio logrando una detección temprana y alerte a los propietarios y se logró evitar posibles pérdidas materiales.

## OBJETIVOS ESPECÍFICOS

- Construcción de prototipo electrónico.
- Monitorear mediante un eco constante del sensor de proximidad con una distancia desde 2cm hasta 400cm.
- Enviar notificación en tiempo real y en caso de hacer detección de algún movimiento.
- Reportar al usuario en tiempo real por medio una salida de información que genere el dispositivo usando una app mediante smtplib (Correo electrónico) en caso de detectar movimiento por rangos establecidos.
- Visualizar mediante una pantalla **oled 1306** la distancia donde se detecta movimiento en el lugar.

## III. METODOLOGÍA DE DESARROLLO

Hard  
ware

# SISTEMA DE PROXIMIDAD

Autor, *Gustavo Alexander Ramos Benavides*

- Módulo WiFi + Bluetooth **ESP32**



Ilustración 1. Fuente: <https://ferretronica.com/>

- Fuente de Alimentación para Protoboard 5V-3.3V (700 mA)



Ilustración 2. Fuente: <https://ferretronica.com/>

- Sensor **hc-sr04**



Ilustración 3. Fuente: <https://ferretronica.com/>

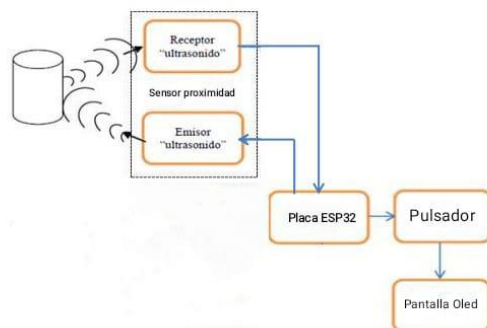
- Pulsador 2 pines
- Módulo Desplaye LCD OLED 128x64 1.3"



Ilustración 4. Fuente: <https://ferretronica.com/>

- Cable Usb

## DIAGRAMA DE BLOQUES



Fuente propia

## DIAGRAMA DE ECO

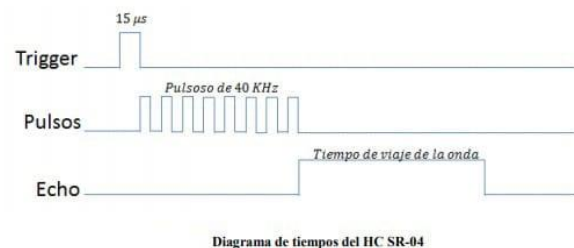


Diagrama de tiempos del HC SR-04

Fuente

# SISTEMA DE PROXIMIDAD

Autor, *Gustavo Alexander Ramos Benavides*

Software

Thonny IDE de Python

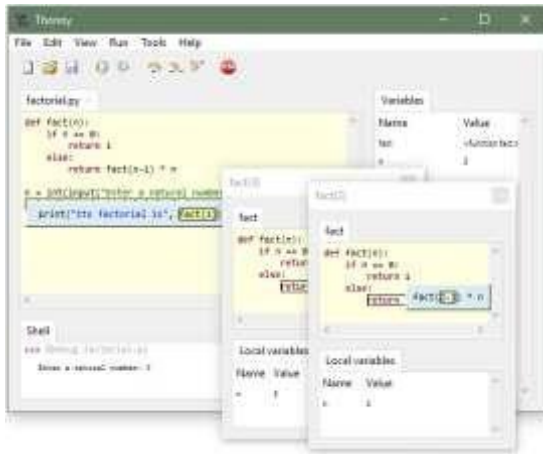


Ilustración 5.

Fácil de empezar. Thonny viene con Python 3.9 incorporado, por lo que solo se necesita un instalador simple y está listo para aprender programación. (También puede usar una instalación de Python independiente, si es necesario). La interfaz de usuario inicial está despojada de todas las características que pueden distraer a los principiantes.

Pythonanywhere

Es una interfaz de línea de comandos Python y Bash basadas en servidor, donde genere una consola para enviar notificaciones a uno o diferentes correos electrónicos, puede enviar datos, crear una visualización instantánea de datos en vivo y enviar alertas.

Librerías Python

El lenguaje Python presenta una sintaxis y semántica referente a la biblioteca estándar. Existen muchos componentes opcionales que se incluyen en las distribuciones de Python y esto le permite al sistema contar con la portabilidad de los programas.

Las librerías Python son amplias, con gran cantidad de producciones en contenidos. Consta de módulos que permiten el acceso de funcionalidades del sistema como entrada y salida de archivos, soluciones estandarizadas a problemas de programación, etc.

Los módulos en Python se encargan de alentar y reforzar la portabilidad de programas que separan especificaciones de la plataforma y conseguir APIs neutrales.

Import SSD1306 es un controlador CMOS OLED / PLED de un solo chip con controlador para el sistema de visualización gráfica de matriz de puntos de diodos emisores de luz orgánicos / poliméricos. SSD1306 consta de 132 segmentos, 64 comunes que pueden admitir una resolución de pantalla máxima de 132 X 64. Está diseñado para paneles OLED de tipo cátodo común.

Módulos Integrados de Python

- Network: Permite la conexión a internet.
- Time: El módulo time (utime) de la librería estándar de MicroPython proporciona un conjunto de funciones para indicar el tiempo que el microcontrolador lleva encendido, medir intervalos de tiempo e introducir retardos (esperas) en la ejecución de los programas.
- Urequests: Requests es una librería para HTTP, ref: licenciada bajo Apache2 <apache2>, escrita en Python, para seres humanos. El módulo urllib2 que se encuentra en el estándar de Python, ofrece la mayoría de las funcionalidades necesarias para HTTP, pero su api está completamente rota. Fue construida para otra época, y una web diferente.
- Framebuf: Las pantallas OLED (Organic light-emitting diode o diodos emisores de luz orgánicos) son una excelente opción para mostrar datos en los proyectos de electrónica.

Son pantallas formadas por diodos (orgánicos) individuales que emiten su propia luz. Esto les permite tener un alto contraste, un gran ángulo de visión y un reducido espesor.

MicroPython dispone de una librería (no estándar) para pantallas monocromas OLED con el controlador SSD1306 y bus de comunicación I2C o SPI.

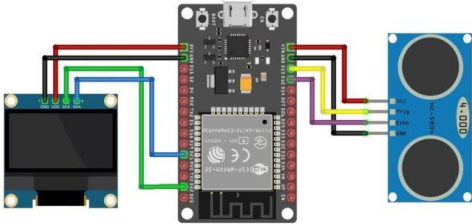
## TRABAJO FUTURO

Este prototipo, puede funcionar a futuro conectando diferentes sensores que generen mejor funcionalidad para el usuario, ejemplo: sensor de movimiento, alarma sonora, este sistema se implementa con batería intenciente por la facilidad en la necesidad de energía baja, no se vería afectado por falta de energía.

# SISTEMA DE PROXIMIDAD

Autor, *Gustavo Alexander Ramos Benavides*

- **hc-sr04** Sensor de proximidad
- **oled 1306** Pantalla
- Pulsador
- I2C: Lectura Oled



## OBTECIÓN DE DATOS

Importación de modulo y librerías:

```
from hcsr04 import HCSR04
from machine import Pin, SoftI2C
import SSD1306, network, time
import network
import urequests
```

Fuente: Propia

Inicialización de librería del Oled:

```
i2c = SoftI2C(scl=Pin(22), sda=Pin(21))
lcd = SSD1306.SSD1306_I2C(128, 64, i2c)
lcd.invert(True)
#coordenadas de la pantalla
cordenadaXTitulo = 30
CordenadaYTitulo = 5
distanciaSensor = 0
```

Fuente: Propia

Lectura del sensor:

```
sensor = HCSR04(trigger_pin=13, echo_pin=12, echo_timeout_us=1000000)
lcd.invert(True)
```

Fuente: Propia

Conexión a wifi:

```
#conexion a wifi
def conexionWiFi(SSID, PASSWORD):
    import network # importa el módulo network
    import urequests
    global sta_if
    sta_if = network.WLAN(network.STA_IF) # instancia el objeto -sta_if- para controlar la interfaz STA
    if not sta_if.isconnected(): # si no existe conexión...
        sta_if.active(True) # activa el interfaz STA del ESP32
        sta_if.connect(SSID, PASSWORD) # inicia la conexión con el AP
        print('Conectando a la red', SSID + "...")
        while not sta_if.isconnected(): # ...si no se ha establecido la conexión...
            pass # ...repite el bucle...
```

Fuente: Propia

Envío de notificación:

```
response = urequests.get("http://leviathandavid.pythonanywhere.com/")
response.close()
print('Configuración de red (IP/netmask/gw/DNS):', sta_if.ifconfig())

#Funcion del envio del correo
def enviarCorreo():
    import urequests
    response = urequests.get("http://alex1423.pythonanywhere.com/")
    response.close()
    print("Movimiento detectado")
```

Fuente propia

Envío de mensaje a pantalla:

```
def imprimirTituloPantalla(texto):
    lcd.text(texto, 30, 5)

def imprimirCuerpoPantalla(texto, cordenadaX, cordenadaY):
    lcd.text(texto, cordenadaX, cordenadaY)
    return;
```

Funcionamiento de todo el sistema:

```
try:
    conexionWiFi("NO DISPONIBLE", "Ds2480748")
    while True:
        lcd.fill(0) # limpia el contenido de la pantalla
        lecturaDistanciaSensorUltrasonico = sensor.distance_cm()
        imprimirTituloPantalla('Mensaje')
        #imprimirCuerpoPantalla(str(lecturaDistanciaSensorUltrasonico), 30, 40)
        if lecturaDistanciaSensorUltrasonico > 5 and lecturaDistanciaSensorUltrasonico <= 30:
            imprimirCuerpoPantalla("Alerta a 30cm", 10, 30)
            imprimirCuerpoPantalla("Alerta email", 10, 45)
            enviarCorreo()
        elif lecturaDistanciaSensorUltrasonico > 40 and lecturaDistanciaSensorUltrasonico <= 70:
            imprimirCuerpoPantalla("Alerta a 50cm", 10, 40)
        elif lecturaDistanciaSensorUltrasonico > 100:
            imprimirCuerpoPantalla("Sin novedad", 10, 40)#Los valores 30, 40 centran texto

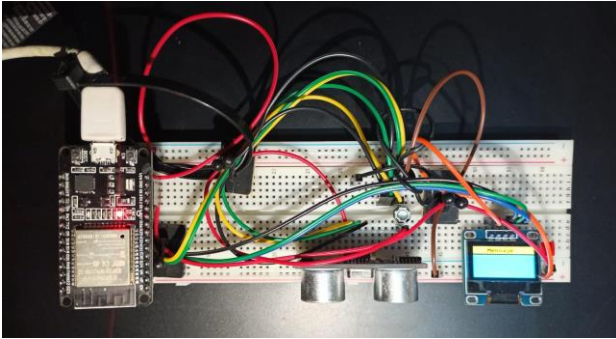
        lcd.show() # muestra o actualiza los valores centrados en la pantalla
```

Fuente propia

# SISTEMA DE PROXIMIDAD

Autor, *Gustavo Alexander Ramos Benavides*

Prototipo:



Fuente propia

Código:

<https://github.com/AlexanderRamosB/Sistema-de-Proximidad>

## CONCLUSIONES

Gracias a Python y su alta maleabilidad y compatibilidad con las tecnologías actuales basados en microcontroladores que son el musculo de todo aparato electrónico en la actualidad; usando esta herramienta tan poderosa en combinación con el microcontrolador ESP32 y sus funcionalidades se logró crear un dispositivo capaz de detectar en el lugar, se crea un prototipo para evidenciar el funcionamiento del sistema de proximidad, Podemos afirmar que el desarrollo del componente expuesto durante el presente documento cumple las expectativas iniciales y objetivos a cabalidad, el monitoreo es continuo, la medición se muestra en la pantalla oled, está información viaja a través un servidor y enviar una notificación correctamente a los correos registrados por el usuario.

El desarrollo e innovación de herramientas que nos brinden seguridad en nuestros hogares o locales comerciales en la actualidad es de gran importancia.

## REFERENCIAS

- [1] **PYTHONANYWHERE** se crea app para el envío de mensajes, <https://www.pythonanywhere.com/>
- [2] **THONNY**, IDE de Python para principiantes

[5] La Biblioteca Estándar de Python — documentación de Python - 3.9.6 <https://docs.python.org/>

[6] MICROPYTHON ESP32 – LIBRERÍA ESTÁNDAR time – ESPloradores <https://www.esploradores.com/time>

## BIBLIOGRAFIA

Información sobre sensor

<https://techtotinkerblogspot.com/2020/10/012-esp32-micropython-hc-sr04.html?m=1>