

Detector de movimiento con Raspberry PI (noviembre del 2024)

John Haider Giraldo, Alejandro Cifuentes,
Wilmer Soto Vidal.

Resumen - El siguiente proyecto es un dispositivo creado utilizando un Raspberry PI, un buzzer, un sensor infrarrojo pasivo (PIR) y un teclado matricial que tiene como propósito ser un sistema detector de movimientos y que posteriormente alerte con un sonido. También tiene un sistema de autenticación para desactivar la alerta, la idea es crear un dispositivo de seguridad para proteger algún bien y/o propiedad.

Índice de Términos - Detector, movimiento, Raspberry Pi, sensores, buzzer, seguridad.

I. INTRODUCCIÓN

Este proyecto se basa en la construcción de un sistema de detección de movimiento usando un Raspberry PI, un sensor infrarrojo, un buzzer y un teclado matricial.

El Raspberry Pi se usa como el controlador principal de todo el sistema, este se programa a través de Python y la conexión entre los diferentes sensores se realiza a través de sus pines GPIO.

Bajo el contexto del curso de Sistemas Operativos este proyecto sirve de gran manera para ilustrar algunos de los conceptos trabajados durante todo el desarrollo de la materia. Se manejan temas como controladores (drivers), manejo de I/O y la gestión de recursos de hardware.

Adicionalmente, este proyecto nos permite experimentar con el tema de sistemas embebidos, también nos deja experimentar con conceptos básicos de electrónica y programación de un Raspberry PI.

Se siguió entonces un video de apoyo [1] el cual nos guía en la construcción del proyecto tanto para la conexión de la Raspberry Pi con los sensores como con el código desarrollado en Python.

Se logró entonces a final de proyecto la correcta implementación de todo, resultando un sensor de detector de movimiento que al activarse emite un sonido de alerta.

También se implementó de manera adicional un teclado matricial el cual funciona con una verificación de clave de 4 caracteres lo que permite al usuario entonces desactivar el sistema de alerta.

El sistema entonces hace registro de cada detección de movimiento y también en caso de que haya autenticación fallida mediante el teclado. Esto nos permite emular un sistema de alarma encontrado en una casa..

II. MARCO TEÓRICO

Para el desarrollo del proyecto se es necesario tener en cuenta los siguientes conceptos:

- **Programación en Raspberry Pi:** Es necesario conocer el uso de lenguajes como Python para interactuar con los pines GPIO y controlar dispositivos externos, como el sensor PIR y el zumbador.
- **Integración con sensores externos:** Comprender cómo los sensores, como el sensor infrarrojo pasivo (PIR), detectan cambios en el entorno físico y comunican esos datos a través de señales digitales que pueden ser procesadas por la Raspberry Pi.
- **Conceptos básicos de electrónica:** Incluir conocimientos sobre cómo conectar y alimentar dispositivos electrónicos externos (como sensores y zumbadores) a través de la Raspberry Pi.

III. METODOLOGÍA

Para la construcción del sistema de alarma con Raspberry Pi, se siguió una metodología dividida en las siguientes etapas:

1. **Definición del problema:**
 - Se identificó la necesidad de desarrollar un sistema de alarma simple y efectivo, capaz de detectar movimiento y emitir una alerta sonora.
2. **Selección de componentes:**
 - Se eligieron los elementos esenciales para el proyecto: una Raspberry Pi como controlador, un sensor PIR para detección de movimiento, un buzzer para emitir alertas y un teclado matricial para la interacción del usuario.
3. **Diseño del sistema:**
 - Se definió la lógica del programa en Python que controla la interacción de los dispositivos conectados a los pines GPIO.
 - Se establecieron los pines GPIO específicos para cada componente según su función.
4. **Configuración del hardware:**
 - Los componentes fueron conectados físicamente a la Raspberry Pi utilizando un protoboard y cables Dupont, siguiendo el esquema descrito en la sección de conexiones físicas.
 -
5. **Desarrollo del software:**
 - Se creó un programa en Python que:

- Monitorea el sensor PIR para detectar movimiento.
- Activa el buzzer como alarma cuando se detecta movimiento.
- Permite al usuario desactivar o activar el sistema mediante un teclado matricial y una clave de 4 dígitos.

6. Pruebas y ajustes:

- Se realizaron pruebas unitarias para validar el funcionamiento de cada componente (sensor PIR, buzzer, teclado).
- Se probó el sistema integrado, simulando escenarios reales para verificar su correcto funcionamiento.
- Se ajustó el programa para corregir errores, como bucles infinitos y mensajes repetitivos.

IV. IMPLEMENTACIÓN

La implementación del sistema se llevó a cabo en dos fases principales: la integración del hardware y el desarrollo del software.

A. Integración del hardware

1. Conexión del sensor PIR:

- El pin de alimentación (VCC) se conectó al pin físico 2 (5V) de la Raspberry Pi.
- El pin de tierra (GND) se conectó al pin físico 6.
- La salida del sensor (OUT) se conectó al pin físico 16 (GPIO 23).

2. Conexión del buzzer:

- El terminal positivo del buzzer se conectó al pin físico 18 (GPIO 24).
- El terminal negativo se conectó a tierra (GND).

3. Conexión del teclado matricial (4x4):

- Las filas (R1, R2, R3, R4) se conectaron a los pines GPIO 21, 20, 16 y 12 respectivamente.
- Las columnas (C1, C2, C3, C4) se conectaron a los pines GPIO 6, 5, 7 y 8.

B. Desarrollo del software

1. Configuración de los pines GPIO:

- Se utilizó la biblioteca **RPi.GPIO** para configurar los pines como entrada o salida según la necesidad.
- Se configuraron resistencias pull-down para estabilizar las señales del teclado matricial.

2. Lógica del programa:

- Se desarrolló un algoritmo en Python que controla el estado del sistema:
 - Activa el buzzer si el sensor PIR detecta movimiento.

- Permite ingresar una clave a través del teclado matricial para desactivar o activar la alarma.
- Muestra mensajes claros al usuario dependiendo del estado del sistema.

3. Monitoreo del sensor PIR:

- Se implementó un hilo separado para monitorear continuamente el estado del sensor PIR sin bloquear la interacción del usuario con el teclado.

4. Control del buzzer:

- El buzzer se activa solo cuando se detecta movimiento y permanece encendido hasta que el usuario desactiva la alarma.

5. Interfaz con el teclado matricial:

- Se diseñó un sistema para detectar qué tecla fue presionada mediante la activación secuencial de las filas y la lectura de las columnas.

6. Mensajes de usuario:

- Se implementaron mensajes claros para indicar al usuario el estado actual del sistema y las acciones requeridas (por ejemplo, "Ingrese la clave para desactivar la alarma").

V. PROTOCOLO DE EXPERIMENTACIÓN

El propósito de estos experimentos es validar la funcionalidad, eficiencia y seguridad del sistema detector de movimientos para así garantizar su correcto funcionamiento.

Entonces para las pruebas del sistema se realizó lo siguiente:

- Se probaron los componentes de forma individual y en conjunto:
 - El sensor PIR para detectar movimiento.
 - El buzzer para emitir alertas.
 - El teclado matricial para ingresar y validar la clave.
 - Se realizaron ajustes al código para mejorar la funcionalidad y la experiencia del usuario.

Entonces se controlaron las siguientes variables para tener un control del funcionamiento:

- Tipo de movimiento (lento, rápido)
- Distancia al sensor (1, 3, 5 metros)
- Ingreso de clave usando el teclado
- Tasa de detecciones exitosas

Se eligen estas tres variables a controlar ya que son las más importantes para determinar el funcionamiento del sistema y cualquier limitación que este tenga.

VI. RESULTADOS

Se realizaron entonces 5 pruebas para cada condición y variable especificadas en el punto anterior.

A. Tipos de movimiento

- Lento
5/5 intentos exitosos
- Rápido
4/5 intentos exitosos

B. Distancia al sensor

- 1 metro
5/5 intentos exitosos
- 3 metros
5/5 intentos exitosos
- 5 metros
5/5 intentos exitosos

C. Ingreso de clave usando el teclado

5/5 intentos exitosos

D. Tasa de detecciones exitosas

Teniendo en cuenta entonces todas las pruebas anteriores tenemos una tasa de detección de:

29/30 o 96,66%

Se encuentra entonces que un punto fallido de nuestro sistema pueden ser los objetos/personas que se muevan de forma rápida ya que el test fallo una vez realizando esta prueba.

VII. CONCLUSIONES

Se realizó entonces de forma exitosa la implementación del sistema detector de movimientos usando Raspberry Pi, sensor PIR, buzzer y teclado matricial.

También demostramos la viabilidad de este sistema de ser una solución eficiente, funcional y de bajo costo para funciones de seguridad.

Se cumplieron todos los objetivos propuestos y se destacan entonces los siguientes puntos clave del sistema:

- Alta tasa de detección
El sensor PIR usado logró detectar movimientos en las 29 de las 30 pruebas realizadas, aunque su precisión disminuye ligeramente al verse al frente de objetos/personas con movimientos rápidos.
- Seguridad en la autenticación
El teclado matricial es eficaz para ejercer un control sobre el sistema y garantizar su funcionamiento como un sistema de alarma.

En resumen, el proyecto presentado es una solución práctica y accesible como sistema de seguridad. Los experimentos realizados y datos adquiridos respaldan la efectividad del sistema.

VIII. REFERENCIAS

[1] Computadoras y Sensores. (2021, 6 marzo). *Raspberry Pi detector de movimiento con sensor infrarrojo pasivo PIR y buzzer, código en Python*. [Video]. YouTube. <https://www.youtube.com/watch?v=HxjvP9m-hRo>