

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela ciencias y Sistemas
Arquitectura de Computadores y Ensambladores 1



PRACTICA 2

Smart Home

No.	Nombre	Carnet
1	Kevin Manuel Veliz Galvez	201901441
2	Mario Alexander Ruano García	201902382
3	Byron Enrique Rumpich Sal	201907769
4	Iris Carolina Paz Guzman	202101728
5	Oscar David Padilla Vásquez	202103250

INTRODUCCIÓN

En este proyecto, se desarrollará un sistema de automatización del hogar utilizando la Raspberry Pi para convertir las viviendas de la residencial "Pinos Altos" en Smart Homes. Se emplearán tecnologías avanzadas como sensores de temperatura, detectores de fuego, botones de acceso y motores para controlar el aire acondicionado, todos programados en Python.

El diseño e implementación de los circuitos electrónicos se realizará directamente en el entorno físico, asegurando la correcta funcionalidad en la práctica. Los pines GPIO de la Raspberry Pi permitirán la comunicación directa con los sensores y actuadores, mientras que una pantalla LCD de 16x2 mostrará información relevante a los usuarios, como el estado de acceso y la temperatura del hogar.

El sistema integrará control manual y automático para dispositivos específicos, mejorando la comodidad y seguridad del hogar. El enfoque se centrará en mantener la seguridad del sistema de control de acceso y en automatizar inteligentemente los dispositivos del hogar, como el aire acondicionado, en función de las condiciones ambientales. No se utilizarán controladores externos o hardware avanzado más allá de los componentes básicos disponibles.

Objetivos Generales

1. Desarrollar un sistema sencillo de domótica que integre varios dispositivos y sensores, mejorando la comodidad y seguridad de las casas en la residencial “Pinos Altos”.
2. Familiarizarse con el uso de la Raspberry Pi Zero para automatizar procesos en el hogar.

Objetivos Específicos

1. Implementar un circuito físico que incluya dispositivos de automatización del hogar, como botones de acceso, sensores de temperatura y detectores de fuego.

Contenido

En el contexto del proyecto de automatización del hogar para la residencial "Pinos Altos", se emplean varios componentes clave que contribuyen al funcionamiento eficiente y seguro del sistema. A continuación, se presentan definiciones y explicaciones de los componentes utilizados, con un enfoque orientado a la arquitectura de computadoras y sistemas embebidos.

Sensor de Flama

Un sensor de flama es un dispositivo electrónico diseñado para detectar la presencia de fuego o llamas. Estos sensores generalmente utilizan fotodiodos o fototransistores para capturar la radiación infrarroja emitida por una llama. La señal detectada es convertida en una señal eléctrica que puede ser procesada por un microcontrolador o una placa de desarrollo como la Raspberry Pi. En el contexto del proyecto, el sensor de flama permite al sistema automatizado detectar condiciones de incendio, mejorando así la seguridad del hogar.

Arquitectura de Computadoras:

El sensor de flama envía señales digitales o analógicas a través de pines GPIO, que son gestionadas por el procesador de la Raspberry Pi. Estas señales se interpretan mediante el software para activar o desactivar otros dispositivos, como el buzzer o la luz de advertencia.

Buzzer

Un buzzer es un dispositivo que emite sonido cuando recibe una señal eléctrica. Existen dos tipos principales de buzzers: piezoeléctricos y electromagnéticos. En el contexto del proyecto, el buzzer piezoeléctrico se utiliza para emitir una alarma sonora cuando el sensor de flama detecta una llama. El buzzer convierte las señales eléctricas en ondas sonoras, proporcionando una indicación audible de una condición crítica.

Arquitectura de Computadoras: El buzzer se controla mediante los pines GPIO de la Raspberry Pi. Una señal lógica de alto o bajo enviada desde la Raspberry Pi activa o desactiva el buzzer, generando sonidos de alerta que se sincronizan con las señales de los sensores y otros componentes del sistema.

MCP3208

El MCP3208 es un convertidor analógico a digital (ADC) de 12 bits con 8 canales de entrada. Este componente es fundamental para la conversión de señales analógicas (como las del sensor de temperatura) en valores digitales que pueden ser procesados por la Raspberry Pi. Permite la lectura precisa de las señales analógicas desde varios sensores, facilitando la integración de datos analógicos en sistemas digitales.

Arquitectura de Computadoras: El MCP3208 se comunica con la Raspberry Pi a través del protocolo SPI (Serial Peripheral Interface). La Raspberry Pi envía comandos de lectura y recibe datos digitales del MCP3208, lo que permite el monitoreo de variables físicas como la temperatura del ambiente. Este ADC es crucial para la conversión de señales en el proceso de adquisición de datos en el sistema.

L293D

El *L293D es un controlador de motor de puente H que permite el control bidireccional de motores de corriente continua. Este componente es esencial para el manejo de actuadores como el motor que simula el aire acondicionado en el proyecto. El L293D puede manejar corrientes relativamente altas y proporciona protección contra sobrecargas y cortocircuitos.

Arquitectura de Computadoras: El L293D recibe señales de control desde los pines GPIO de la Raspberry Pi, que determinan la dirección y la velocidad del motor. El controlador de motor gestiona las señales eléctricas para dirigir el motor, permitiendo la automatización del control del aire acondicionado basado en las lecturas de temperatura. Esto demuestra cómo un controlador de motor se integra en un sistema embebido para realizar tareas físicas de manera eficiente.

Sensor de Luz : Un sensor de luz es un dispositivo que detecta la intensidad de la luz en su entorno y convierte esta información en una señal eléctrica, que puede ser procesada por un sistema computacional. En la arquitectura de computadoras, estos sensores se utilizan para diversas aplicaciones, como ajustar el brillo de una pantalla, encender o apagar luces automáticamente, o en sistemas más complejos como cámaras y robots autónomos.

Código Comentado

```
import RPi.GPIO as GPIO
from time import sleep
import drivers
from datetime import datetime
import Adafruit_DHT
```

Este bloque importa las librerías necesarias para trabajar con los pines GPIO de la Raspberry Pi, controlar la pantalla LCD, manejar el tiempo, obtener la fecha y hora actual, y leer datos de un sensor de temperatura y humedad DHT11.

```
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
```

Este bloque configura los pines GPIO para que no muestren advertencias y establece el modo de numeración BCM, que se basa en los números de los pines GPIO en la placa Raspberry Pi.

```

# Pines de botones para el ingreso del patrón
BTN_1 = 4
BTN_2 = 17
BTN_3 = 27
BTN_4 = 22

GPIO.setup(BTN_1, GPIO.IN)
GPIO.setup(BTN_2, GPIO.IN)
GPIO.setup(BTN_3, GPIO.IN)
GPIO.setup(BTN_4, GPIO.IN)

# Pines de otros dispositivos
TEMP_SENSOR_PIN = 23
TEMP_CONTROL_PIN = 24
PIR_PIN = 5
FLAME_SENSOR_PIN = 6
BUZZER_PIN = 12

# Nuevos pines ajustados para el segundo script
SENSOR_PIN = 20      # Reasignado desde el pin 4
LED1_PIN = 21        # Reasignado desde el pin 17
BTN_PIN = 19         # Reasignado desde el pin 27

GPIO.setup(TEMP_CONTROL_PIN, GPIO.OUT)
GPIO.setup(PIR_PIN, GPIO.IN)
GPIO.setup(FLAME_SENSOR_PIN, GPIO.IN)
GPIO.setup(BUZZER_PIN, GPIO.OUT)
GPIO.setup(LED_PIN, GPIO.OUT)      "LED_PIN" no está definido
GPIO.setup(LED_CONTROL_PIN, GPIO.OUT) "LED_CONTROL_PIN" no está definido

```

Este bloque define los pines GPIO que se utilizarán para botones, sensores y otros dispositivos como un zumbador y LEDs. Luego, configura cada pin como entrada (**GPIO.IN**) o salida (**GPIO.OUT**).

```

# Configuración de los pines del segundo script
GPIO.setup(SENSOR_PIN, GPIO.IN)
GPIO.setup(LED1_PIN, GPIO.OUT)
GPIO.setup(BTN_PIN, GPIO.IN)

```

Este bloque configura los pines adicionales para sensores y LEDs, reasignando los pines GPIO utilizados previamente en otro script.

```
# Variables de control
patron = 'B03'
patron_guardado = ''
contador = 0
```

Este bloque define las variables que controlan el proceso de ingreso del patrón, incluyendo el patrón esperado, el patrón ingresado por el usuario y un contador para los intentos fallidos.

```
# Paso 1: Ingreso de patrón
print("Inicio")
lcd lcd_display_string("Bienvenido", 1)
lcd lcd_display_string("Ingresar Patron", 2)

while True:
    boton_1 = GPIO.input(BTN_1)
    boton_2 = GPIO.input(BTN_2)
    boton_3 = GPIO.input(BTN_3)
    boton_4 = GPIO.input(BTN_4)

    if boton_1 == 0:
        lcd lcd_clear()
        print("B")
        patron_guardado += 'B'
        lcd lcd_display_string(patron_guardado, 2)
        sleep(0.15)
    if boton_2 == 0:
        lcd lcd_clear()
        print("0")
        patron_guardado += '0'
        lcd lcd_display_string(patron_guardado, 2)
        sleep(0.15)
    if boton_3 == 0:
        lcd lcd_clear()
        print("3")
        patron_guardado += '3'
        lcd lcd_display_string(patron_guardado, 2)
        sleep(0.15)
    if boton_4 == 0:
        print("ENTER")
        lcd lcd_display_string(patron_guardado, 1)
        sleep(0.15)
        if patron_guardado == patron:
```

Este bloque permite al usuario ingresar un patrón utilizando botones físicos.

El patrón se muestra en la pantalla LCD y se compara con el patrón predefinido. Si se ingresa incorrectamente tres veces, el usuario debe esperar antes de volver a intentarlo.


```
# Paso 2: Funcionalidades adicionales
lcd lcd_display_string("Bienvenido", 1)
lcd lcd_display_string("A tu casa :D", 2)
sleep(1)
lcd lcd_clear()

sensor = Adafruit_DHT.DHT11
```

Este bloque muestra un mensaje de bienvenida después de que el usuario ingresa correctamente el patrón y prepara el sensor DHT11 para leer temperatura y humedad.

```
try:
    while True:
        # Mostrar la hora en la pantalla LCD
        lcd lcd_display_string("Horario", 1)
        lcd lcd_display_string(str(datetime.now().time()), 2)

        # Leer temperatura y humedad
        humedad, temperatura = Adafruit_DHT.read_retry(sensor, TEMP_SENSOR_PIN)
        if humedad is not None and temperatura is not None:
            lcd lcd_clear()
            lcd lcd_display_string(f"Temp:{temperatura:.1f}C", 1)
            lcd lcd_display_string(f"Humedad:{humedad:.1f}%", 2)
            if temperatura > 27:
                GPIO.output(TEMP_CONTROL_PIN, GPIO.HIGH)
            else:
                GPIO.output(TEMP_CONTROL_PIN, GPIO.LOW)
        else:
            lcd lcd_display_string("Error en sensor", 1)

        # Detección de movimiento con sensor PIR
        if GPIO.input(PIR_PIN):
            print("Movimiento detectado")
        else:
            print("No hay movimiento")

        # Detección de llama con el sensor de flama
        flame_state = GPIO.input(FLAME_SENSOR_PIN)
        if flame_state == GPIO.LOW:
            print("No hay flama, no se detecta fuego.")
            GPIO.output(BUZZER_PIN, GPIO.HIGH) # Apaga el zumbador
        else:
            print("Flama presente, fuego detectado.")
            GPIO.output(BUZZER_PIN, GPIO.LOW) # Enciende el zumbador
```

Este es el bucle principal que realiza diversas tareas:

- Muestra la hora actual en la pantalla LCD.
- Lee la temperatura y la humedad del sensor DHT11 y controla un dispositivo de acuerdo a la temperatura.
- Detecta movimiento utilizando un sensor PIR y muestra el estado en la consola.
- Detecta llamas utilizando un sensor de flama y activa un zumbador si se detecta fuego.
- Controla un LED basado en la entrada de un sensor y un botón adicional.

```
except KeyboardInterrupt:
    pass

finally:
    GPIO.cleanup() # Limpia la configuración de los GPIO al finalizar
```

Este bloque asegura que si el programa es interrumpido manualmente (por ejemplo, con **Ctrl+C**), los pines GPIO se limpian y restablecen a su estado original.

Presupuesto y Facturas

TETTS /
ENLACANDO TECNOLOGIAS CREATIVAS

TETTS A

NIT: 88938877
Sede: TETTS A S.A.
2da AVENIDA 13 OF. ZONA 9 LOCAL 1-3
CDNF. CONDOMINIO PLAZA 2-11,
Ciudad Guatemala
Sitio web: tettsa.gt
Vendedor: Edwin Rubén Pérez González
Orden de Venta: 520.166

FACTURA

Serie: FEL85538CA
Número: FEL3565765017
Autorización:
B55538CA-0489-4199-B9C2-
1671C6FBC98B
Fecha v.Hora Certificación:
2024-08-19 16:43:27
Nº de Documento
INV1/2024/10525

NIT: CF

Cliente: CONSUMIDOR FINAL
Dirección: CIUDAD

Código	Cant.	Precio	Subtotal
		18.00	18.00
MODULO SENSOR DE FLAMA NO FLAMA			
TOTAL			18.00 Q
Efectivo			20.00
CAMBIO			2.00 Q
IVA			1.93
Total de Impuestos			1.93 Q

TETTS A S.A

Sujeto a pagos trimestrales ISR

SELLO DE PAGO

TETTS A

TECNOLOGIA Y ELECTRONICA S.A.

CANCELADO

VECINOS

LIBRERIA Y PAPELERIA

4ta Avenida 1-13 Zona 3
Chimaltenango, Chimaltenango.

CONSUMIDOR FINAL

Ciudad

Vendedor: MARILYN
NIT: 8264091-2

NIT: CF

Numero Interno: 3655

FACTURA ELECTRONICA

Serie: 00332817
Numero: No.: 1479427153
Fecha: 13/08/2024 12:06:17

CANTIDAD	DESCRIPCION	PRECIO U	TOTAL
1	CARTULINA BOWEN (TEXCOTE 14)	Q5.00	Q5.00
1	CARTON ILUSTRACION/PRESENTACION COLORES 32X20"	Q28.50	Q28.50
3	SILICON BARRA DELGADA TRANSPARENTE	Q1.50	Q4.50

TOTAL EN LETRAS TREINTA Y OCHO QUETZALES EXACTOS

Sujeto a pagos trimestrales ISR

TOTAL: Q38.00

DOCUMENTO TRIBUTARIO ELECTRONICO

Numero de Autorización: 00332817-582E-4451-A627-90887054EF78
CERTIFICADOR: DIGIFACT, S.A. NIT Certificador: 77454820
Fecha de Certificación: 13/08/2024 12:06:20

fel

Factura Electrónica

Factura

SILVIA LUCRECIA AVILA SATZ DE BAY

NIT Emisor: 7978759

FERRETERIA INTERAMERICANA

3 CALLE 3-13 zona 1, Chimaltenango, CHIMALTENANGO

NIT Receptor: CF

Nombre Receptor: CONSUMIDOR FINAL

NÚMERO DE AUTORIZACIÓN:

0705D82E-80BF-4B84-A12E-773D36A44C70

Serie: 0705D82E Número de DTE: 2378124164

Número Acceso:

Fecha y hora de emisión: 13-ago-2024 12:27:33

Fecha y hora de certificación: 13-ago-2024 00:27:33

Moneda: GTQ

#No	B/S	Cantidad	Descripción	Precio/Valor unitario (Q)	Descuentos (Q)	Otros Descuentos(Q)	Total (Q)	Impuestos
1	Bien	1	Pieza Darpapel 1/4	25.00	0.00	0.00	25.00 IVA	2.67871
TOTALES:					0.00	0.00	25.00 IVA	2.67871

* Sujeto a pagos trimestrales ISR

Datos del certificador

Superintendencia de Administración Tributaria NIT: 16693949

2 kjs acetato ——— 2.-

2 pliegos papel ——— 2.50

Ø 4.50

LIBRERIA Y PAPELERIA

SANTA MARIA

18 CALLE 3-45 ZONA 4, CHIMALTENANGO

gas

COMERCIAL F&CH S.A.

TEL.: +502 24767800 / +502 4977168 / +502 3849 5859

http://www.festinterior.com

tienda@fchinterior.com

tienda de interior, no es factura.

Servido por Yonier Gonzalez

NO-PRO60 modulo sensor

botonesistencia 3 pines

3.3V 1 Unidad(es) x 15.50

15.50

CB-204F cable alupont

20cm 4 unidades F-F

5 Unidad(es) x 3.50

17.50

NO-1212 Modulo zumbador

buzzer activo

1 Unidad(es) x 12.00

12.00

LD-510 LED 5mm Diapico

blanco

8 Unidad(es) x 1.00

8.00

TOTAL

53.00

Efectivo (GTQ)

55.00

CAMBIO

2.00

IVA por Pagar

5.69

Total de Impuestos

5.69

Gracias por su compra, vuelva pronto.

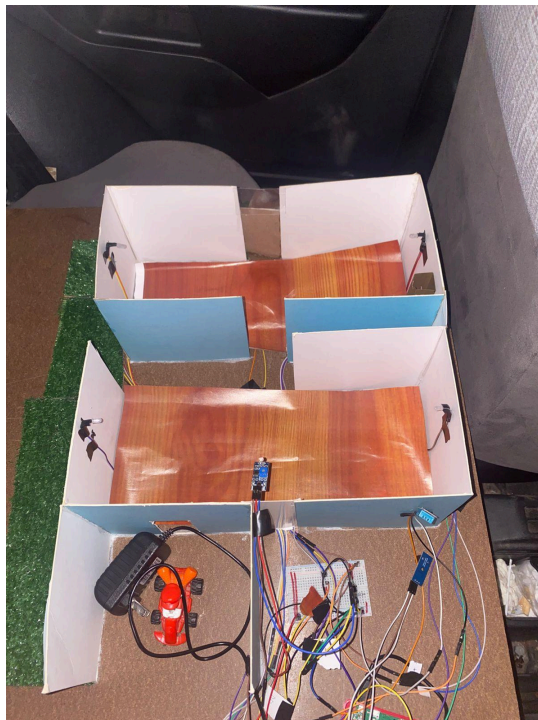
Pedido 02968-002-002

19/8/2024, 9:35:19

Conclusiones

1. **Detección y Seguridad:** El sensor de flama es crucial para la detección temprana de incendios, mejorando la seguridad del hogar al enviar señales a la Raspberry Pi, que posteriormente activa alarmas sonoras mediante un buzzer. Esto asegura una respuesta rápida ante condiciones de peligro.
2. **Conversión de Señales y Procesamiento de Datos:** El MCP3208 es fundamental en la conversión de señales analógicas, como las provenientes de un sensor de temperatura, en datos digitales procesables por la Raspberry Pi. Esta capacidad de conversión es vital para integrar sensores analógicos en un sistema digital, permitiendo un monitoreo preciso de variables físicas.
3. **Control de Actuadores:** El L293D permite el control eficiente de motores de corriente continua, como los utilizados en sistemas de aire acondicionado, mediante la gestión de señales eléctricas desde la Raspberry Pi. Esto demuestra cómo los componentes de control, junto con la lógica de sistemas integrados, pueden automatizar tareas físicas basadas en datos de sensores.
4. **Arquitectura de Computadoras:** La interacción entre los diversos componentes y la Raspberry Pi a través de pines GPIO y protocolos como SPI, resalta la importancia de la arquitectura de computadoras en la implementación de sistemas embebidos para automatización. La gestión precisa de señales y la sincronización entre los componentes son clave para el funcionamiento integral del sistema automatizado.

Imágenes de la maqueta y componentes



Maqueta Física

Sensores y Componentes

