

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



PROYECTO

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 los últimos años, el desarrollo de tecnologías orientadas a la automatización del hogar ha avanzado de manera significativa, dando lugar a lo que hoy conocemos como "Smart Home" o hogares inteligentes. Estos sistemas permiten a los usuarios controlar y monitorear diversos dispositivos y funciones dentro de una casa mediante sensores, actuadores y software especializado, todo ello con el fin de mejorar la comodidad, seguridad y eficiencia energética. El acceso a componentes asequibles y la integración de tecnologías de Internet de las Cosas (IoT, por sus siglas en inglés) han facilitado la creación de hogares inteligentes, donde la interacción entre el entorno y el usuario es cada vez más sofisticada.

Este proyecto tiene como objetivo principal el desarrollo de un sistema de automatización para la residencia "Pinos Altos", que desea convertir algunas de sus viviendas en hogares inteligentes. Para ello, se integrarán diversas tecnologías y componentes, utilizando una Raspberry Pi y otros dispositivos electrónicos como sensores, actuadores y sistemas de comunicación. El sistema permitirá el control remoto de las funciones del hogar a través de una API y una aplicación web, facilitando la interacción con los diferentes dispositivos desde cualquier lugar.

Objetivos Generales

1. Desarrollar un proyecto funcional de IoT utilizando una Raspberry Pi, que permita la automatización y control remoto de las funciones de una vivienda inteligente.

Objetivos Específicos

1. Comprender el funcionamiento de las entradas y salidas digitales de la Raspberry Pi para su uso en un entorno de IoT.
2. Aplicar conocimientos de programación en Python para implementar estructuras de control en la Raspberry Pi.
3. Implementar diversos tipos de comunicación entre dispositivos mediante la Raspberry Pi y otras plataformas, como Arduino.
4. Analizar y aplicar el uso de la Raspberry Pi en el contexto de hogares inteligentes en el mundo real.
5. Integrar dispositivos adicionales como sensores, actuadores y sistemas de comunicación, para ampliar las funcionalidades del sistema de domótica.

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

Este bloque importa los módulos y bibliotecas necesarias. Algunas de estas son: Flask: Para crear una API que gestione peticiones HTTP. requests: Para realizar solicitudes HTTP a servicios externos. Flask-CORS: Manejo de la política de CORS en la API. RPi.GPIO: Permite el control de los pines GPIO en Raspberry Pi. Adafruit_DHT: Control de sensores de temperatura y humedad. drivers: Controlador de pantalla LCD. threading: Para ejecutar procesos en paralelo. signal, sys: Para capturar señales del sistema y realizar la limpieza al finalizar el proceso.

```
import random
from flask import Flask, request, jsonify
import requests
from flask_cors import CORS
import datetime
import time
import threading
import RPi.GPIO as GPIO
from time import sleep
import drivers
from datetime import datetime
import Adafruit_DHT
import signal
import sys
```

Aquí se inicia el servidor Flask que permitirá recibir solicitudes HTTP, mientras que CORS permite que la API sea accesible desde otras fuentes externas (importante en sistemas web distribuidos).

```
app = Flask(__name__)
CORS(app)
```

Aquí se realiza la configuración de los pines GPIO. Se establece el modo de numeración de los pines y se configuran los pines para diferentes dispositivos (botones, sensores, LEDs, buzzer). Se definen pines tanto para entrada (lectura de datos) como para salida (control de dispositivos externos).

```
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)

TEMP_SENSOR_PIN = 23
TEMP_CONTROL_PIN = 24
PIR_PIN = 5
FLAME_SENSOR_PIN = 6
BUZZER_PIN = 12
LED_CONTROL_PIN = 13

SENSOR_PIN = 20
LED1_PIN = 21
BTN_PIN = 19

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_CONTROL_PIN, GPIO.OUT)
GPIO.setup(SENSOR_PIN, GPIO.IN)
```

Se definen variables globales para manejar el estado del sistema y el patrón de acceso que se introducirá mediante botones. Además, se establecen URL y claves de API para realizar solicitudes HTTP a una base de datos MongoDB externa.

```
lcd = drivers.Lcd()
patron = '803'
patron_guardado = ''
contador = 0

URL_API = "https://us-east-1.amazonaws.com/data-mongodb-api.com/app/data-uvdudur/eng
URL_KEY = "Lm7aHb14scG8LedryL9NK7NP7kwb4bZQ29DThf3nufw141gxFQuqw3IG5qDx7M"
```

La ruta /control permite recibir comandos para controlar dispositivos como luces, garaje, aire acondicionado y alarmas. Dependiendo de los datos recibidos (como el estado de 'on' o 'off'), se ejecutarán diferentes acciones sobre los pines GPIO.

```
@app.route("/control", methods=['POST'])
def control():
    data = request.json

    if 'light' in data:
        # lógica para controlar la luz
        pass

    if 'garage' in data:
        # lógica para controlar la puerta del garaje
        pass

    if 'air' in data:
        # lógica para controlar el aire acondicionado
        pass

    if 'alarm' in data:
        # lógica para controlar la alarma
        pass

    print("Datos recibidos:", data)

    return jsonify({ 'ejecutado': True })
```

Esta función envía información a una base de datos externa, como acciones de encendido o apagado de dispositivos. Se utiliza la API de MongoDB para registrar las acciones realizadas y la fecha correspondiente.

```
def enviarInfo(accion, on_off):
    path = "/insertOne"
    url = URL_API + path
    output_date = datetime.datetime.now().strftime("%Y-%m-%dT%H:%M:%S.00Z")
    headers = {'api-key': URL_KEY, 'Accept': 'application/json'}
    data = {
        "collection": "registros",
        "database": "smarthome",
        "dataSource": "proyectoArqui",
        "document": {"accion": accion, "on_off": on_off, "Fecha_registro":
    }
    response = requests.post(url, json=data, headers=headers)
    print(response.json())
```


Esta función ejecuta un bucle infinito que:

Muestra mensajes en la pantalla LCD.

Gestiona el patrón de acceso introducido por el usuario.

Monitorea sensores como el de movimiento (PIR), el sensor de flama, y el sensor de temperatura y humedad.

Controla dispositivos como el aire acondicionado, luces y alarmas según las lecturas de los sensores.

Esta función se ejecuta en un hilo paralelo, lo que permite que el sistema funcione de forma continua mientras el servidor Flask gestiona solicitudes.

```
def bucle_infinito():  
    print("Inicio")  
    lcd lcd_display_string("Bienvenido", 1)  
    lcd lcd_display_string("Ingresar Patron", 2)  
  
    # Lógica para ingresar patrón mediante botones  
    # Lógica para manejar sensores, temperatura, y más  
    # Control de dispositivos según el estado de los sensores
```

Este bloque se encarga de capturar señales de interrupción (Ctrl+C) y realizar una limpieza de los GPIO al finalizar la ejecución, garantizando que los recursos de hardware se liberen correctamente.

```
def shutdown_handler(signal_received, frame):  
    print("La API se está deteniendo...")  
    GPIO.cleanup() # Limpia la configuración de los GPIO  
    sys.exit(0) # Finaliza el programa
```

```
if __name__ == '__main__':  
    app.run(debug=True)
```

Finalmente, este bloque asegura que el servidor Flask se ejecute si el script se ejecuta directamente. Se establece en modo debug para facilitar la depuración durante el desarrollo.

Presupuesto y Facturas

TETTS /
ENLACANDO TECNOLOGIAS CREATIVAS

TETTS A

NIT: 88938877

VENEDOR: TETTS A S A

404 AVENIDA 13 OF ZONA 9 LOCAL 1-3

CDNF CONDOMINIO PLAZA 2-11,

Ciudad Guatemala

Sitio web: tetsa.gt

Vendedor: Edwin Ruben Perez Gonzalez

Orden de Venta: 520166

FACTURA

Serie: FEL85538CA

Numero: 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 ELECTRONICOS 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

Fecha de Certificación 13/08/2024 12:06:20

NIT Certificador: 77454820

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
1a. CALLE 3-45 ZONA 4, CHIMALTENANGO

SM

gas

COMERCIALIZADORA RACH, S.A.
Tel.: +502 24767800 / +502 4977168 / +502 3849 5895
http://www.rach.com.gt
tira@rach.com.gt
Tirador de Entrega, no es factura.
Servido por Yonier González

NO-PRO60 modulo sensor
fotonesistencia 3 pines
3.3V 1 Unidad(es) x 15.50 15.50
CB-204F cable adapt
20cm 4 unidades F-F
5 Unidad(es) x 3.50 17.50
MO-1212 Modulo zumbador
buzzer activo
1 Unidad(es) x 12.00 12.00
LD-510 LED 5mm Diapico
difuso empaquetado
blanco
8 Unidad(es) x 1.00 8.00

TOTAL 53.00

8.00

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



Fronted

LOGIN

SMART HOME #3

DashboardControl

Iniciar Sesión

Control

Usuario

Contraseña

Ingresar

Al ingresar tendrás acceso a todos los componentes de la casa.

Control y mensajes

SMART HOME #3

DashboardControl

Estado de los componentes

Temperatura de la casa73°C

Luces de la casa

Aire acondicionado

Alarma de incendio

Porton del garage

LCD

Ver mensaje en la LCD15s

Mensaje a mostrar

Enviar

Notificaciones

9/21/24, 9:12 AM
Mensaje de la alarma

9/21/24, 9:12 AM
Mensaje de la alarma

9/21/24, 9:12 AM
Mensaje de la alarma

9/21/24, 9:12 AM
Mensaje de la alarma

9/21/24, 9:12 AM
Mensaje de la alarma

Graficas y Datos

