

Informe Final - Sistema de Domótica Hospitalaria

Angie Marisela García Ríos
Estefany Cuervo Suárez
Juan Felipe Arroyave Zapata
Electrónica Digital II

Índice

1. Identificación del proyecto	3
2. Resumen	3
3. Descripción del hardware	3
4. Descripción del software	3
5. Arquitectura del sistema	4
5.1. Sensores implementados (4)	4
5.2. Actuadores implementados (4)	4
5.3. Comunicaciones	4
6. Funciones principales del sistema	4
6.1. Monitorización en tiempo real	4
6.2. Sistema de alarmas	5
6.3. Control del entorno	5
6.4. Interfaz de usuario	5
7. Explicación de funciones principales	5
8. Flujo de operación	6
9. Umbrales médicos configurados	6
10. Características de seguridad	6
11. Pruebas realizadas	7
12. Consideraciones técnicas	7
12.1. Limitaciones	7
12.2. Mejoras futuras	7

13.Entregables técnicos	8
14.Conclusión	10

1. Identificación del proyecto

- **Nombre del proyecto:** Sistema de Domótica Hospitalaria para Habitación de Paciente
- **Integrantes del equipo:** Angie Marisela García Ríos, Estefany Cuervo Suárez, Juan Felipe Arroyave Zapata
- **Fecha:** 26 de noviembre de 2025

2. Resumen

Este proyecto consiste en el diseño e implementación de un sistema de domótica hospitalaria para la monitorización y control automatizado de una habitación de paciente. El sistema integra sensores médicos y ambientales, actuadores para el control del entorno, comunicación WiFi, interfaz web local, y un bot de Telegram para notificaciones y control remoto. El objetivo es mejorar la seguridad, eficiencia y trazabilidad en el cuidado de pacientes.

3. Descripción del hardware

- **ESP32-C3** (microcontrolador principal)
- **Sensor DHT11** (temperatura y humedad) conectado al pin 0
- **Servo SG90** para control de cortinas (pin 20)
- **Servo SG92R** para control de puertas (pin 4)
- **Buzzer pasivo** para alarmas sonoras (pin 1)
- **LED** para simular iluminación (pin 2)
- **LED indicador WiFi** (pin 5)
- **Protoboard y cableado** para conexiones

4. Descripción del software

- **Lenguaje:** MicroPython
- **Librerías:** network, urequests, machine, dht, time, json
- **IDE:** Thonny
- **Protocolos:** HTTP, API REST

5. Arquitectura del sistema

5.1. Sensores implementados (4)

- **DHT11:** Temperatura y humedad ambiental
- **MAX30100 (simulado):** Frecuencia cardíaca y SpO2
- **Sensor de posición de cortina**
- **Sensor de posición de puerta**

5.2. Actuadores implementados (4)

- **Servo motor SG90:** Control de cortinas
- **Servo motor SG92R:** Control de puertas
- **Buzzer:** Alarmas sonoras
- **LED:** Iluminación de la habitación

5.3. Comunicaciones

- **WiFi:** Conexión a red local
- **Telegram Bot:** Notificaciones y control remoto
- **ThingSpeak:** Almacenamiento y visualización de datos en la nube
- **API REST:** Comunicación con servicios web

6. Funciones principales del sistema

6.1. Monitorización en tiempo real

- Temperatura y humedad ambiental
- Frecuencia cardíaca (simulada)
- Nivel de oxígeno en sangre SpO2 (simulado)
- Estado de cortinas y puertas
- Estado de iluminación

6.2. Sistema de alarmas

- Alarma por frecuencia cardíaca: ¡50 bpm o >120 bpm
- Alarma por SpO2 bajo: <90 %
- Alarma por temperatura alta: >30°C
- Alarma por humedad alta: >70 %
- Notificaciones automáticas por Telegram
- Alarma sonora intermitente

6.3. Control del entorno

- Apertura/cierre automático de cortinas
- Control de puertas
- Gestión de iluminación
- Activación/desactivación manual de alarmas

6.4. Interfaz de usuario

- Bot de Telegram con comandos:
 - /start - Menú de inicio
 - /estado - Estado completo del sistema
 - /datos - Datos médicos en tiempo real
 - /cortina_abrir/cerrar - Control de cortinas
 - /puerta_abrir/cerrar - Control de puertas
 - /luz_encender/apagar - Control de iluminación
 - /alarma_silencio - Silenciar alarmas
- ThingSpeak: Dashboard en la nube para visualización de datos

7. Explicación de funciones principales

- conectar_wifi(): Gestión de conexión WiFi con indicador LED
- leer_sensores(): Lectura de todos los sensores del sistema
- verificar_alarmas(): Evaluación de condiciones de alarma
- procesar_comandos_telegram(): Manejo de comandos vía Telegram

- `enviar_thinkspeak()`: Envío de datos a la nube
- `activar_alarma_sonora()`: Control de alarmas audibles
- `mover_servo_sg90()`, `mover_servo_sg92r()`: Control de actuadores

8. Flujo de operación

1. Inicialización de hardware y conexión WiFi
2. Verificación del bot de Telegram
3. Configuración inicial de actuadores
4. Bucle principal cada 100ms:
 - Lectura de sensores cada 3 segundos
 - Verificación de alarmas
 - Procesamiento de comandos Telegram cada 3 segundos
 - Envío a ThingSpeak cada 30 segundos
 - Control intermitente de alarma sonora

9. Umbrales médicos configurados

- **Frecuencia cardíaca:** 50-120 bpm
- **SpO2:** >90 %
- **Temperatura:** >30°C
- **Humedad:** >70 %

10. Características de seguridad

- Sistema de offset en Telegram para evitar procesamiento duplicado
- Manejo de excepciones en todas las funciones críticas
- Reconexión automática WiFi
- Timeouts en solicitudes HTTP
- Validación de estados del sistema

11. Pruebas realizadas

1. Conexión WiFi y indicador LED
2. Lectura correcta de sensores DHT11
3. Simulación de datos médicos con variaciones realistas
4. Activación/desactivación de alarmas por umbrales
5. Control de actuadores (servos, LED, buzzer)
6. Comunicación con Telegram (envío y recepción)
7. Envío de datos a ThingSpeak
8. Manejo de errores y recuperación

12. Consideraciones técnicas

12.1. Limitaciones

- Sensor MAX30100 simulado (en implementación real requeriría hardware específico)
- Dependencia de conexión WiFi estable
- Limitaciones de memoria en ESP32-C3
- Timeout de Telegram puede afectar respuestas en redes lentas

12.2. Mejoras futuras

- Implementación de sensor MAX30100 real
- Interfaz web local integrada
- Base de datos local para histórico de datos
- Múltiples habitaciones con coordinación central
- Protocolo MQTT para mejor escalabilidad

13. Entregables técnicos

- Código fuente completo en MicroPython
- Documentación de conexiones hardware
- Configuración de servicios cloud (ThingSpeak, Telegram)
- Manual de usuario para operación del sistema
- Evidencias de pruebas realizadas



Logo y nombre del proyecto



Habitacion del paciente



Baño del paciente

14. Conclusión

El sistema de domótica hospitalaria implementado demuestra la viabilidad de integrar tecnologías IoT en entornos médicos para mejorar la monitorización de pacientes y la eficiencia operativa. La combinación de sensores médicos, actuadores ambientales y comunicaciones inalámbricas proporciona una solución robusta y escalable para la automatización de habitaciones hospitalarias.