



SISTEMAS ELECTRÓNICOS DIGITALES **PROYECTO DOMÓTICA**

CURSO 2023/2024

EUGENIO FERNÁNDEZ HIDALGO - 55837
PEDRO ÁLVAREZ MONTEAGUDO – 55717
DIEGO RAMÍREZ FUENTE – 56049

GRUPO A404

Índice

1.	INTRODUCCIÓN	3
2.	Herramientas	4
3.	Estructura y Funcionamiento.....	6
4.	Conclusiones y Futuros Proyectos	9

1. INTRODUCCIÓN

La domótica, referida a la automatización de las funciones de una casa, representa un campo en constante crecimiento. La programación en C para microprocesadores es una herramienta clave en este sector, permitiendo el control y la automatización de dispositivos y sistemas dentro del hogar. En este proyecto, se desarrolló un sistema que utiliza un microprocesador para controlar distintas funciones de una casa inteligente, como la iluminación y la seguridad.



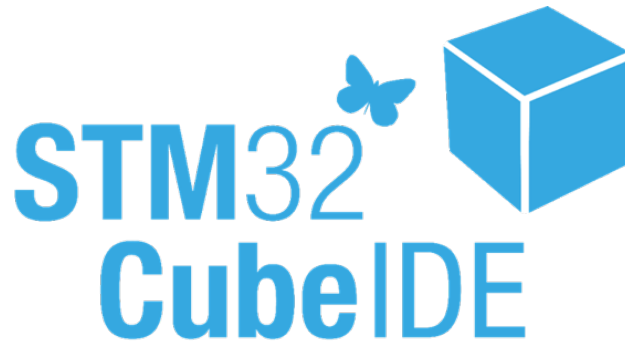
En la última década, la domótica ha experimentado un crecimiento exponencial, impulsada por avances en IoT y microelectrónica. La integración de microprocesadores en sistemas domóticos permite una gestión más inteligente y adaptable de los dispositivos del hogar. Este proyecto explora el uso de un microcontrolador STM32F411 para automatizar aspectos críticos de una casa, como la iluminación y la seguridad, en función de las condiciones ambientales y la presencia de personas.



El código desarrollado en C para este proyecto es fundamental para el control y la lógica del sistema. C, conocido por su eficiencia y bajo nivel de abstracción, es ideal para trabajar con hardware y sistemas embebidos como los microcontroladores STM32. Este proyecto representa un ejemplo práctico de cómo la programación en C se puede aplicar para mejorar y automatizar las funciones del hogar.

2. Herramientas

STM32CubeIDE es una herramienta integral que combina un editor de código, compilador, y herramientas de depuración, específicamente diseñada para simplificar el desarrollo de aplicaciones basadas en microcontroladores STM32.



El microcontrolador STM32F411 se seleccionó por su equilibrio entre rendimiento, coste y facilidad de uso. Además, el proyecto incorporó componentes electrónicos clave como LEDs, que sirven como indicadores visuales, un sensor de ultrasonidos HC-SR04 para detectar la presencia y distancia, un sensor de infrarrojos para detectar movimiento y un LDR para medir la intensidad de la luz.

La elección de estos componentes se basó en su capacidad para simular un entorno domótico realista, en el que la luz, la presencia y la distancia son parámetros críticos. La integración de estos elementos en el proyecto demuestra cómo los microcontroladores pueden interactuar con una variedad de sensores y actuadores para crear un sistema domótico funcional.

Los sensores utilizados en este proyecto son fundamentales para interactuar con el entorno físico. Incluyen:

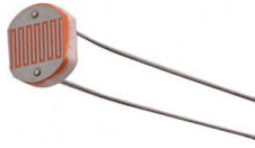
- Sensor de ultrasonidos HC-SR04: Utilizado para medir distancias mediante ultrasonidos. Esencial para detectar la presencia y el movimiento en áreas específicas, como el portal de la casa.



- Sensor de infrarrojos: Permite detectar movimiento y presencia sin contacto físico, utilizado en este proyecto para la activación de la luz del portal.



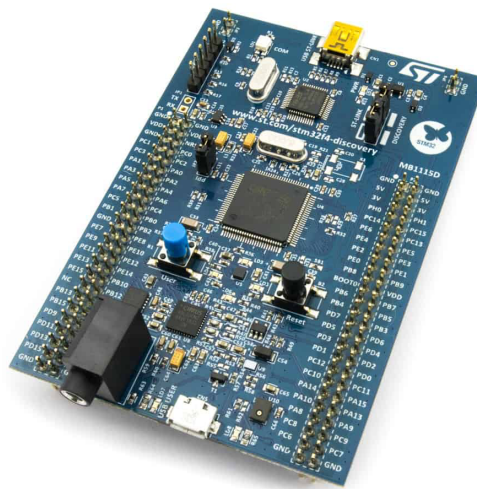
- LDR (Resistor Dependiente de Luz): Este sensor analógico detecta la intensidad de la luz. Se utiliza para controlar la iluminación exterior, encendiéndose automáticamente cuando oscurece.



Cada uno de estos sensores aporta una dimensión crucial al sistema de domótica, permitiendo que el microcontrolador STM32F411 responda de manera inteligente a las condiciones ambientales y a la presencia de personas.

El STM32F411 destaca por su equilibrio óptimo entre rendimiento, coste y facilidad de uso. Sus características incluyen:

- Alta capacidad de procesamiento: Ideal para manejar múltiples entradas y salidas simultáneamente.
- Bajo consumo de energía: Clave para aplicaciones domóticas donde la eficiencia energética es importante.
- Flexibilidad en la conectividad: Permite la integración fácil con una variedad de sensores y actuadores.
- Compatibilidad con STM32CubeIDE: Facilita el desarrollo y depuración del código.



La elección de este microcontrolador es fundamental para garantizar la eficiencia y fiabilidad del sistema domótico.

A continuación, se presenta una imagen de la maqueta realizada:

El programa principal en C, ejecutado en el microcontrolador STM32F411, está estructurado para monitorear constantemente las entradas de los sensores y responder de manera adecuada. Utiliza interrupciones para manejar eventos en tiempo real, como la detección de movimiento o cambios en la iluminación, lo que permite una respuesta rápida y eficiente del sistema. La conversión AD/DC es esencial para procesar las señales analógicas de los sensores, como el LDR, transformándolas en valores digitales que el microcontrolador puede utilizar.

El código gestiona varios elementos clave:

- La función 'activa_alarma()' controla el LED de alarma, activándolo durante un período específico cuando se detecta presencia.
- La función 'sensor_distancia()' maneja el sensor de ultrasonidos HC-SR04, encendiendo un LED si un objeto se encuentra a una distancia determinada.
- La función 'HAL_ADC_ConvCpltCallback()' gestiona la conversión AD y la lectura de los valores de los sensores.

Las interrupciones y temporizadores se utilizan para medir la distancia y gestionar la temporización de la alarma.

Estas funciones demuestran la capacidad del sistema para interactuar con el entorno físico, realizar mediciones precisas y tomar decisiones en función de esos datos.

El código del proyecto, escrito en C, es el núcleo del sistema domótico. Algunas características y funciones clave incluyen:

- Uso del DMA (Direct Memory Access): Permite la transferencia eficiente de datos entre la memoria y los periféricos sin la intervención de la CPU, crucial para leer los sensores de manera eficiente.
- Gestión de interrupciones: El código utiliza interrupciones para responder rápidamente a eventos como la detección de movimiento o cambios en la iluminación.
- Funciones específicas: Por ejemplo, la función 'activa_alarma()' controla el LED de alarma, activándolo durante un período específico cuando se detecta presencia.

El siguiente fragmento del código que ilustra el uso del DMA y la gestión de interrupciones:

```
// Inicialización y configuración del DMA
HAL_ADC_Start_DMA(&hadc1, adc_value, 2);

// Función de interrupción para la conversión AD completa
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc) {
    dma_completo = 1; // Finaliza la lectura DMA
}
```

A continuación se detalla como se ha trabajado con los distintos sensores:

Sensor de Ultrasonidos.

Código:

```
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim) {
    if (htim->Channel == HAL_TIM_ACTIVE_CHANNEL_1) {
        if (es_pv == 0) {
            // Si es el flanco de subida, capturamos el valor actual del contador
            valor1 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_1);
            es_pv = 1;
            // Cambiamos la polaridad de captura para esperar el flanco de bajada
            __HAL_TIM_SET_CAPTUREPOLARITY(htim, TIM_CHANNEL_1, TIM_INPUTCHANNELPOLARITY_FALLING);
        } else if (es_pv == 1) {
            // Si es el flanco de bajada, capturamos el valor actual del contador
            valor2 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_1);
            // Reiniciamos el contador para medir el siguiente pulso
            __HAL_TIM_SET_COUNTER(htim, 0);

            // Calculamos el período del pulso (tiempo entre flanco de subida y bajada)
            if (valor2 > valor1) {
                periodo = valor2 - valor1;
            }

            // Reiniciamos la bandera de flanco de subida
            es_pv = 0;
            // Cambiamos la polaridad de captura para esperar el próximo flanco de subida
            __HAL_TIM_SET_CAPTUREPOLARITY(htim, TIM_CHANNEL_1, TIM_INPUTCHANNELPOLARITY_RISING);
        }
    }
}
```

Durante la interrupción del temporizador, se capturan los valores de tiempo en los flancos de subida y bajada del pulso ultrasónico. Estos valores se utilizan para calcular el período del pulso, que se relaciona con la distancia al objeto. La implementación incluye cambios de polaridad para gestionar los flancos de subida y bajada, permitiendo así una medición precisa del tiempo de vuelo del ultrasonido. Este enfoque es esencial para obtener datos fiables sobre la distancia entre el sensor y los objetos en su entorno.

Sensor LDR.

La lectura del LDR se realiza utilizando un convertidor analógico a digital (ADC) con Direct Memory Access (DMA). A continuación, se presenta un párrafo explicativo del tratamiento del LDR en el código:


```
// Inicialización de variables y funciones
uint32_t adc_value[2]; // Almacena los valores de los sensores analógicos LDR y temperatura
uint32_t ldr_value = 0; // Variable que almacena la lectura del LDR

volatile uint8_t dma_completo = 0; // Bandera para indicar la finalización de la lectura DMA

// Callback llamado cuando la conversión ADC se completa
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc) {
    dma_completo = 1; // Finaliza la lectura DMA
}

// Función para encender un LED basado en el valor del LDR
int encender_led(uint32_t value) {
    if (value < 300) return 1; // Enciende el LED si la lectura del LDR es baja
    else return 0;
}
```

Sensor Infrarrojos.

Código:

```
if (HAL_GPIO_ReadPin(GPIOA, INFRARROJOS_Pin) == GPIO_PIN_SET) {
    // Realiza acciones cuando se detecta el sensor infrarrojo
    // Puedes agregar aquí el código correspondiente a tus necesidades
}
```

En esta sección, se verifica si el pin asociado al sensor infrarrojo (INFRARROJOS_Pin) está en estado alto (GPIO_PIN_SET). Si es así, significa que el sensor infrarrojo ha detectado algo. Dentro de este bloque condicional, puedes agregar el código correspondiente a las acciones que desees realizar cuando el sensor infrarrojo detecta algo.

4. Conclusiones y Futuros Proyectos

El proyecto logró implementar un sistema de control domótico efectivo, demostrando la versatilidad y la potencia del microcontrolador STM32F411 en aplicaciones de domótica. Como futura mejora, se propone la integración de un sensor de temperatura, preparado en el diseño actual pero no implementado. Dicho añadido, ya contemplado en el diseño, podría ofrecer un control más completo sobre el ambiente del hogar, ajustando automáticamente sistemas de calefacción o refrigeración; proporcionando una automatización aún más completa de la casa.

Otras mejoras podrían incluir la conexión a Internet para el control remoto y la monitorización, la integración con otros dispositivos domóticos y la utilización de algoritmos más avanzados para el procesamiento de datos y la toma de decisiones.

Este proyecto ha demostrado con éxito la aplicación de microprocesadores en sistemas domóticos para mejorar la seguridad y la comodidad en el hogar. A través del uso eficiente de sensores y la programación en C, se ha creado un sistema que responde de manera inteligente a las condiciones ambientales y la presencia humana. Las posibilidades de expansión y mejora son amplias.