



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA Y DISEÑO INDUSTRIAL
Ronda de Valencia, 3 - 28012 Madrid

SISTEMAS ELECTRÓNICOS DIGITALES:

APLICACIÓN DOMÓTICA CON MICROCONTROLADOR

Miembros:

Iker Castiella Aguirrezabala (55791)

Sonia Carrero Díaz (55784)

1 INTRODUCCIÓN	2
2 COMPONENTES	3
2.1 POTENCIÓMETRO	3
2.2 SERVOMOTOR	3
2.3 BOTÓN	4
2.4 LEDS	5
2.5 LDR	5
2.6 PLACA PROTOBOARD	6
2.7 SENSOR INFRARROJO	6
3 FUNCIONAMIENTO	7
4 PROGRAMACIÓN	8
3.1 STM32CUBE	8
3.2 CONFIGURACIÓN DE PERIFÉRICOS	8
3.3 MAIN del programa	8
3.4 OTRAS CONFIGURACIONES	11
5 MAQUETA	13
6 CONCLUSIONES	14
7 ENLACES	14

1 | INTRODUCCIÓN

En el ámbito de la electrónica digital, un microprocesador es el núcleo principal de numerosos sistemas electrónicos, encargados de ejecutar instrucciones y procesar datos de manera eficiente. Estos dispositivos son fundamentales en el desarrollo de soluciones tecnológicas avanzadas, incluyendo aplicaciones domóticas que buscan mejorar la automatización y la interacción en los hogares modernos.

En este trabajo, se explora la implementación de un sistema domótico utilizando el microcontrolador STM32F411E-DISCO. Este microcontrolador específico ofrece un entorno de desarrollo robusto y versátil, diseñado para aplicaciones que requieren alta eficiencia energética y rendimiento optimizado. Además, se emplea STM32CubeIDE, un entorno integrado de desarrollo que facilita la programación y configuración del microcontrolador, aprovechando al máximo sus capacidades.

El proyecto se centra en la simulación y puesta en práctica de un sistema domótico utilizando una maqueta equipada con los sensores y actuadores necesarios para gestionar diferentes aspectos. Esta representación permite evaluar en un entorno controlado la funcionalidad y fiabilidad del sistema antes de su implementación final en un entorno real.

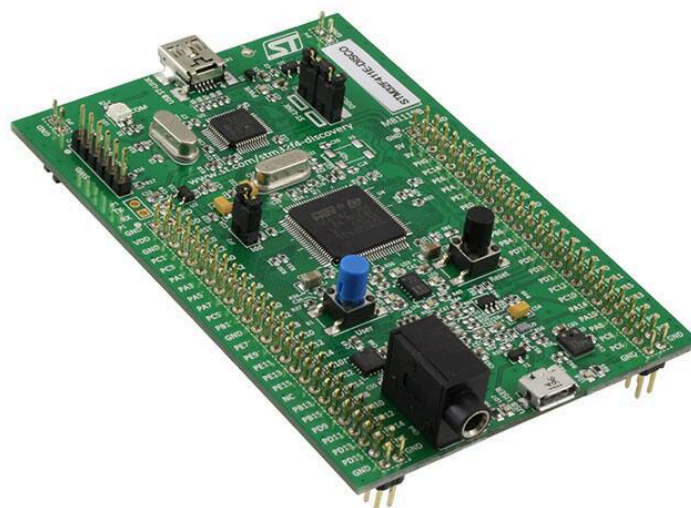


Figura 1: STM32F411E-DISCO

2| COMPONENTES

2.1 POTENCIÓMETRO

Un potenciómetro es un componente electrónico pasivo que actúa como un resistor variable. Consiste en una resistencia fija y un contacto móvil (cursor) que desliza a lo largo de esta resistencia. Al mover el cursor, se cambia la resistencia efectiva entre los terminales del potenciómetro, permitiendo ajustar el nivel de tensión o corriente en un circuito.

El potenciómetro tiene tres terminales: dos terminales fijos conectados a los extremos de la resistencia y un terminal central conectado al cursor. Al variar la posición del cursor, se modifica la relación de resistencias en los dos segmentos de la resistencia total, permitiendo así un control preciso de la señal.



Figura 2: Potenciómetro

2.2 SERVOMOTOR

El servomotor SG90 es un actuador rotatorio de tamaño compacto y bajo costo, ampliamente utilizado en aplicaciones de robótica, modelismo y proyectos de automatización. Este dispositivo está diseñado para ofrecer control preciso de la posición angular mediante señales PWM (modulación por ancho de pulso).

El SG90 consta de un motor de corriente continua (DC), un conjunto de engranajes que reducen la velocidad y aumentan el par, y un circuito de control integrado que interpreta las señales PWM. Estas señales determinan la posición del eje del servomotor, que puede moverse a un ángulo específico dentro de su rango operativo, típicamente de 0 a 180 grados.

El servomotor SG90 tiene tres cables: uno para la alimentación (rojo), otro para tierra (marrón y el tercero para la señal de control (naranja). Cuando se le envía una señal PWM con un ancho de pulso específico, el circuito de control interno posiciona el eje del motor en la correspondiente posición angular.



Figura 3: Servomotor SG90

2.3 BOTÓN

Un botón es un dispositivo de entrada simple que se utiliza para enviar una señal a un circuito electrónico cuando se presiona. Los botones son componentes esenciales en una amplia gama de aplicaciones, desde electrodomésticos hasta sistemas informáticos y proyectos de electrónica.

Cuando se presiona el actuador del botón, los contactos eléctricos internos se cierran (o abren, dependiendo del diseño), completando un circuito y permitiendo que la corriente fluya. Al liberar el botón, el muelle interno devuelve el actuador a su posición inicial, y los contactos eléctricos vuelven a su estado original, interrumpiendo el circuito.



Figura 4: Botón

2.4 LEDS

Los LEDs (diodos emisores de luz) son componentes electrónicos que emiten luz cuando una corriente eléctrica pasa a través de ellos. Son ampliamente utilizados como indicadores visuales en diversos dispositivos y aplicaciones debido a su bajo consumo de energía, larga vida útil y rápida respuesta. En el contexto de un sistema domótico, los LEDs de colores pueden ser muy útiles para indicar el estado de una puerta, facilitando al usuario la visualización rápida de su estado (abierta o cerrada).



Figura 5: LED

2.5 LDR

Un LDR, o resistor dependiente de la luz (también conocido como fotorresistor o fotoconductor), es un tipo de resistor cuyo valor de resistencia cambia en función de la cantidad de luz incidente sobre su superficie. Los LDRs son dispositivos pasivos ampliamente utilizados en aplicaciones donde se requiere detectar la presencia o intensidad de la luz.

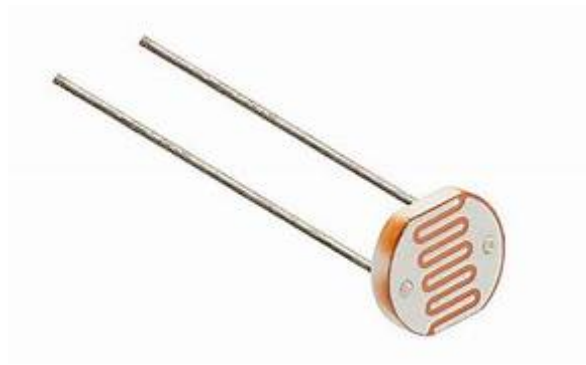


Figura 6: LDR

2.6 PLACA PROTOBOARD

Es una herramienta esencial para la creación y prueba de circuitos electrónicos sin necesidad de realizar soldaduras. Facilita el diseño y la experimentación rápida de circuitos, permitiendo conexiones temporales y fáciles de modificar.

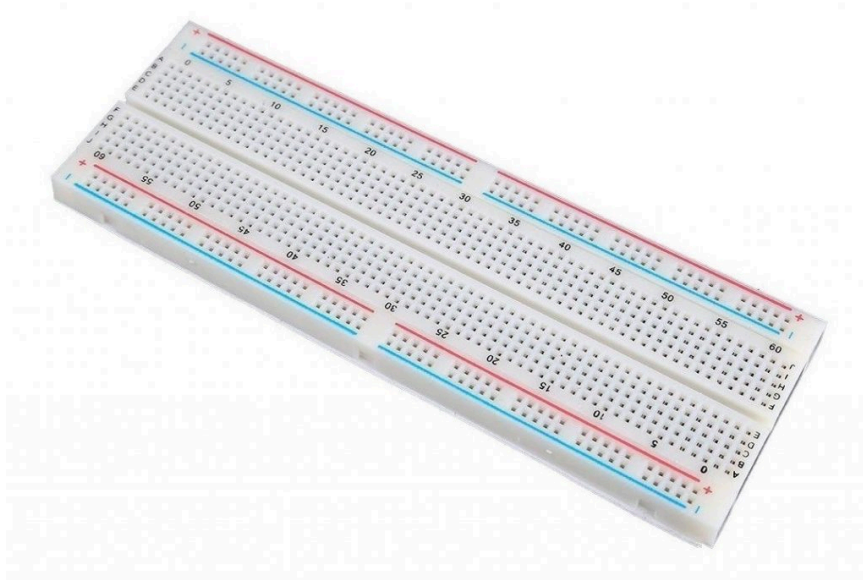


Figura 7: Protoboard

2.7 SENSOR INFRARROJO

Un sensor infrarrojo es un dispositivo electrónico que se utiliza para detectar la radiación infrarroja (IR) emitida por objetos en su entorno. Estos sensores son ampliamente utilizados en aplicaciones de detección de proximidad, medición de distancia, y en sistemas de seguridad y automatización.



Figura 8: Infrarrojo

3| FUNCIONAMIENTO

El objetivo de este proyecto es diseñar un sistema de control, tanto manual como automático, para la puerta de entrada de un supermercado, así como para la iluminación de la entrada.

Primero, las luces en la maqueta están compuestas por LEDs que se encienden cuando no detectan un rayo de luz incidente en el LDR al que están conectados.

Segundo, el servomotor es responsable de abrir y cerrar la puerta de entrada del supermercado. Por defecto, opera de manera automática, girando 90° cuando el sensor infrarrojo instalado en la entrada detecta una presencia. Una vez que la presencia deja de ser detectada, el servomotor retorna la puerta a su posición inicial, cerrándola e impidiendo el paso. Para cambiar al modo de control manual, es necesario pulsar un botón, lo cual desactiva el sensor infrarrojo. En este modo, el control de la puerta se realiza mediante un potenciómetro manejado por el usuario.

En ambos modos de operación, el proyecto incluye dos LEDs: uno verde que indica cuando la puerta está abierta y otro rojo que indica cuando la puerta está cerrada. Estos LEDs permiten a los usuarios conocer el estado de la puerta en todo momento.



Figura 9: ejemplo de puerta automática

4| PROGRAMACIÓN

3.1 STM32CUBE

STM32Cube es un conjunto completo de herramientas y software desarrollado por STMicroelectronics para facilitar el desarrollo y la implementación de aplicaciones en microcontroladores STM32. STM32Cube incluye varias herramientas y recursos que simplifican el proceso de desarrollo, desde la configuración del hardware hasta la escritura y depuración del código.

3.2 CONFIGURACIÓN DE PERIFÉRICOS

GPIO

Pin Name	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/Pull...	Maximum output ...	User Label
PA5	n/a	n/a	External Interrupt ...	No pull-up and no...	n/a	Boton
PA6	n/a	n/a	Input mode	No pull-up and no...	n/a	sensor_infrarrojo
PC4	n/a	Low	Output Push Pull	No pull-up and no...	Low	Trigger
PD12	n/a	Low	Output Push Pull	No pull-up and no...	Low	Farola
PE13	n/a	Low	Output Push Pull	No pull-up and no...	Low	Led_verde
PE15	n/a	Low	Output Push Pull	No pull-up and no...	Low	Led_rojo

ADC

Pin Name	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/Pull...	Maximum output ...	User Label
PA1	ADC1_IN1	n/a	Analog mode	No pull-up and no...	n/a	Potenciometro
PA3	ADC1_IN3	n/a	Analog mode	No pull-up and no...	n/a	LDR

TIM

Pin Name	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/Pull...	Maximum output ...	User Label
PE9	TIM1_CH1	n/a	Alternate Functio...	No pull-up and no...	Low	Servomotor

3.3 MAIN del programa

```
//VARIABLES
volatile uint8_t modo_servo=0; //0-> automatico, 1-> con servomotor
volatile uint8_t cerrar_puerta=0;
uint32_t valor_potenciometro=0;
uint32_t valor_LDR=0;
uint32_t valor_adc[2];
volatile uint8_t DMA_completado = 0;
```

```

//FUNCIONES

//lectura del boton que varia el modo del servomotor mediante interrupciones
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){
    if(GPIO_Pin==GPIO_PIN_5){
        if(modoservo==1)modoservo=0;
        else if(modoservo==0)modoservo=1;
    }
}

//en funcion del valor de luminosidad detectado por el LDR enciende o apaga la farola
int encender_farola(uint32_t value){
    if(value>3000)return 0;
    else return 1;
}

//control del servomotor en modo automatico. Al recibir un valor alto del sensor infrarrojo
//genera un pulso PWM. La lectura se hace mediante un codigo antirrebotes para evitar falsos positivos
void servo_automatico(){
    int contador=0,tiempo=0;

    if((HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_6))&&(cerrar_puerta==0)){
        tiempo=HAL_GetTick();
        while(contador<5){
            if((HAL_GetTick()-tiempo)>=25){
                if(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_6)==0){
                    contador=0;
                    break;
                }
                else {
                    tiempo=HAL_GetTick();
                    contador++;
                }
            }
            if((contador==5)&&(HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_6)==1)){
                HAL_TIM_SET_COMPARE(&htim1,TIM_CHANNEL_1,3);
                HAL_GPIO_WritePin(GPIOE,GPIO_PIN_13,1);
                HAL_GPIO_WritePin(GPIOE,GPIO_PIN_15,0);
                cerrar_puerta=1;
            }
        }
    }
    contador=0;
}

//recibe el valor del potenciómetro y genera un pulso PWM
uint32_t servo_manual(uint32_t value){
    if(value>0 && value<3500){
        HAL_GPIO_WritePin(GPIOE,GPIO_PIN_13,1);
        HAL_GPIO_WritePin(GPIOE,GPIO_PIN_15,0);
        return 3;
    }
    else if(value>=3500){
        HAL_GPIO_WritePin(GPIOE,GPIO_PIN_13,0);
        HAL_GPIO_WritePin(GPIOE,GPIO_PIN_15,1);
        return 9;
    }
    else return 9;
}

//informa que el DMA se ha completado
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc)
{
    DMA_completado = 1;
}

```

```

int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_ADC1_Init();
    MX_TIM1_Init();
    /* USER CODE BEGIN 2 */
    HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_1);
    __HAL_TIM_SET_COMPARE(&htim1,TIM_CHANNEL_1,0);

    HAL_ADC_Start_DMA(&hadc1,valor_adc,2);

    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    int tiempo_puerta=0;
    int aux=0;
    while (1)
    {
        valor_potenciometro=valor_adc[0];
        valor_LDR=valor_adc[1];

        if(modo_servo==1) __HAL_TIM_SET_COMPARE(&htim1,TIM_CHANNEL_1,servo_manual(valor_potenciometro));
        else servo_automatico();

        HAL_GPIO_WritePin(GPIOD,GPIO_PIN_12,encender_farola(valor_LDR));

        //acciona el trigger
        HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,GPIO_PIN_SET);
        HAL_Delay(10);
        HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,GPIO_PIN_RESET);

        if((cerrar_puerta==1)&&(aux==0)){
            tiempo_puerta=HAL_GetTick();
            aux=1;
        }

        if(HAL_GetTick()-tiempo_puerta>2000){
            __HAL_TIM_SET_COMPARE(&htim1,TIM_CHANNEL_1,9);
            HAL_GPIO_WritePin(GPIOE,GPIO_PIN_13,0);
            HAL_GPIO_WritePin(GPIOE,GPIO_PIN_15,1);
            cerrar_puerta=0;
            aux=0;
        }

    }

    /* USER CODE END WHILE */

```

La frecuencia del reloj es de 100 Mhz



Para las entradas analógicas hemos escogido 12 bits (15 ADC Clock cycles)

ADC1 Mode and Configuration		
Mode		
<input type="checkbox"/>	IN0	
<input checked="" type="checkbox"/>	IN1	
<input type="checkbox"/>	IN2	
<input checked="" type="checkbox"/>	IN3	
▼	ADCs_Common_Settings	
	Mode	Independent mode
▼	ADC_Settings	
	Clock Prescaler	PCLK2 divided by 2
	Resolution	12 bits (15 ADC Clock cycles)
	Data Alignment	Right alignment
	Scan Conversion Mode	Enabled
	Continuous Conversion Mode	Enabled
	Discontinuous Conversion Mode	Disabled
	DMA Continuous Requests	Enabled
	End Of Conversion Selection	EOC flag at the end of single channel conversion
▼	ADC_Regular_ConversionMode	
	Number Of Conversion	2
	External Trigger Conversion Source	Regular Conversion launched by software
	External Trigger Conversion Edge	None
>	Rank	1
>	Rank	2

Figura 12: configuración ADC (potenciómetro y LDR)

El servomotor funciona a 50 Hz y con un ancho de pulso de entre 1 y 2 milisegundos

▼	Counter Settings	
	Prescaler (PSC - 16 bits value)	20000-1
	Counter Mode	Up
	Counter Period (AutoReload Register - 16 bits val..	100-1
	Internal Clock Division (CKD)	No Division
	Repetition Counter (RCR - 8 bits value)	0
	auto-reload preload	Enable
▼	Trigger Output (TRGO) Parameters	
	Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
	Trigger Event Selection	Reset (UG bit from TIMx_EGR)
▼	Break And Dead Time management - BRK Configurati..	
	BRK State	Disable
	BRK Polarity	High
▼	Break And Dead Time management - Output Configura..	
	Automatic Output State	Disable
	Off State Selection for Run Mode (OSSR)	Disable
	Off State Selection for Idle Mode (OSSI)	Disable
	Lock Configuration	Off

Figura 13: configuración TIM1

5| MAQUETA

Se ha realizado el local del supermercado, teniendo la posibilidad de quitar el techo para que quede más accesible al control de la puerta manual.

El círculo del fondo simula el día y la noche. En función del lado de la semicircunferencia que muestre, una linterna apuntará al LDR para activar las luces de la entrada o no.

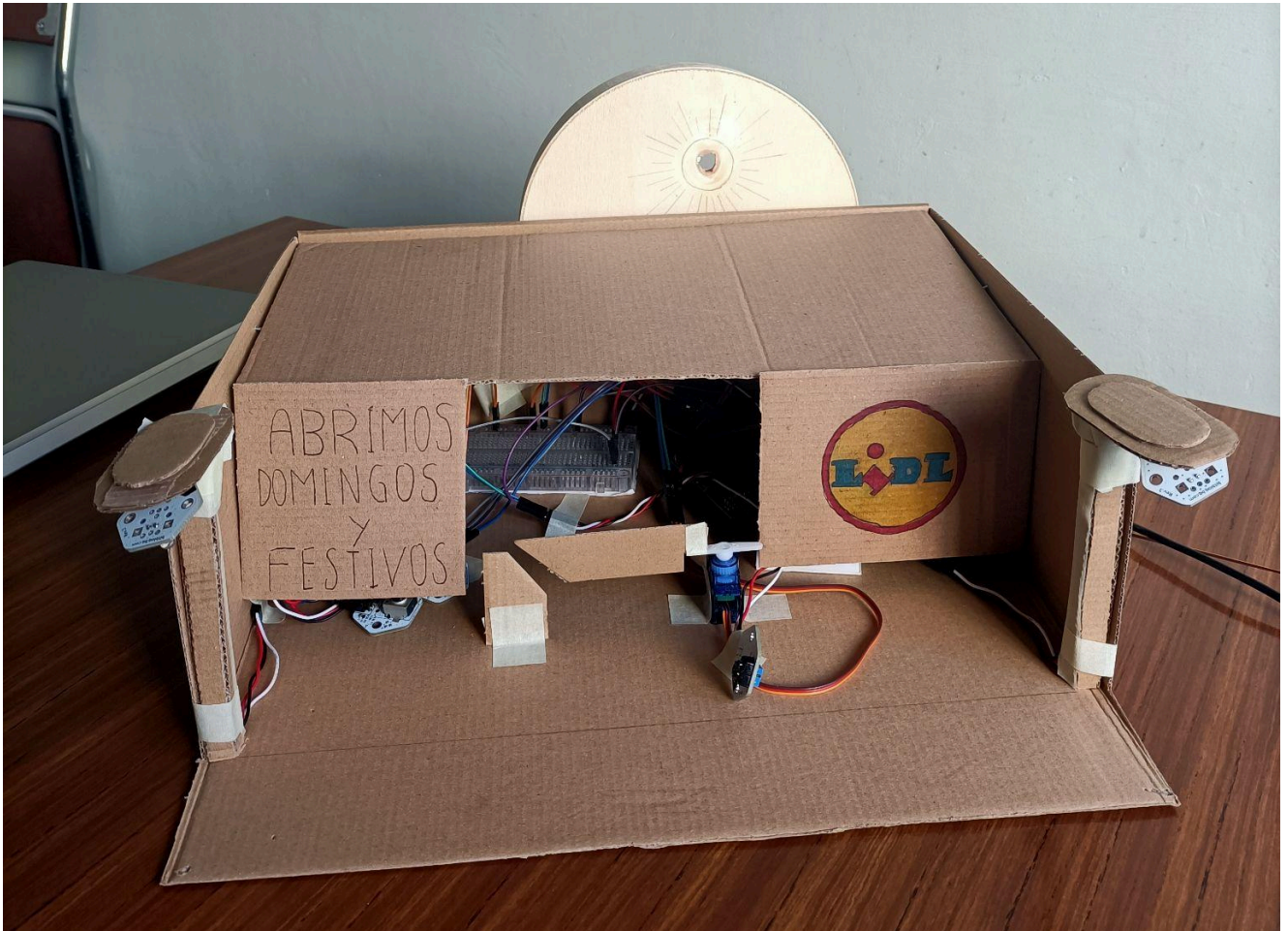


Figura 14: exterior de la maqueta

En la siguiente foto se ven los LEDs que indican el estado de la puerta junto al servomotor acompañado del sensor de infrarrojos. Al otro lado de la puerta se observa el botón encargado de la modalidad de la puerta y del potenciómetro para controlar el modo manual.

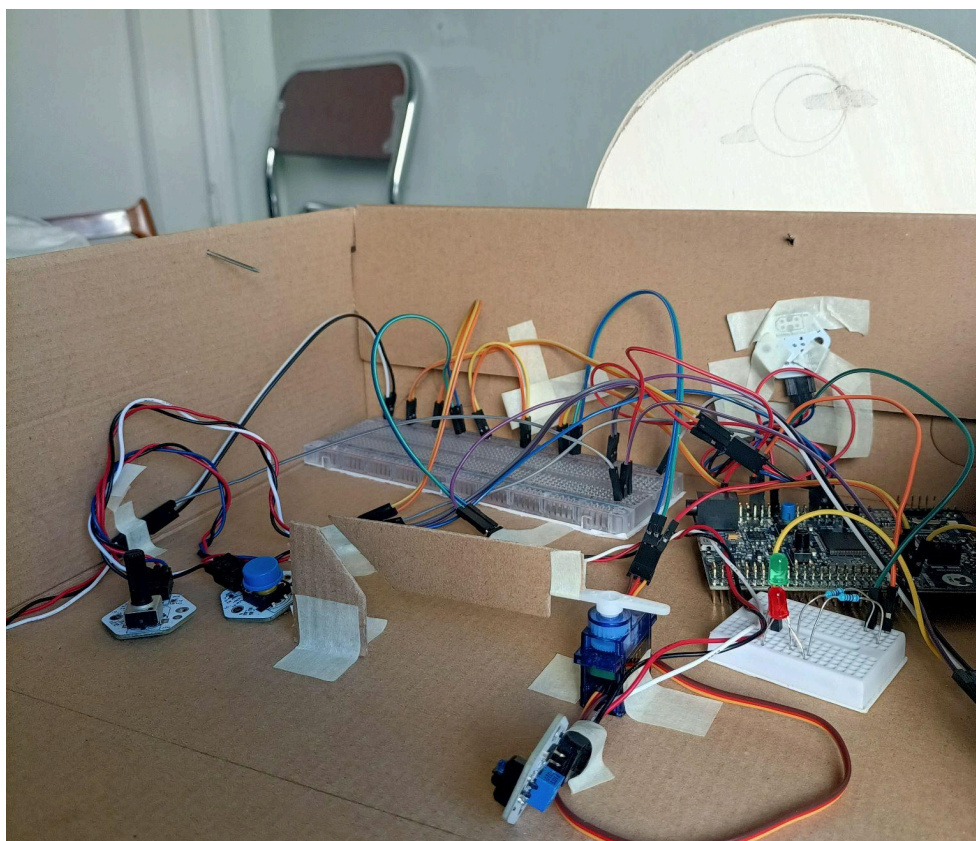


Figura 15: interior de la maqueta

6| CONCLUSIONES

A lo largo de este proyecto, hemos ampliado y aplicado nuestros conocimientos sobre el programa STM32Cube, llevándolos a la práctica en un entorno real. La implementación del sistema de control tanto manual como automático para la puerta de entrada y las luces de un supermercado nos ha permitido explorar diversas funcionalidades y superar retos técnicos.

En el transcurso del desarrollo, encontramos algunos inconvenientes, especialmente en la configuración de las lecturas de dos canales analógicos: el potenciómetro y el LDR. Sin embargo, con persistencia y ajustes adecuados, logramos resolver estos problemas y obtener lecturas precisas y confiables de ambos dispositivos.

Asimismo, enfrentamos dificultades con el sensor infrarrojo, ya que inicialmente detectaba la luz del día como una presencia, lo que interfiere con su funcionamiento correcto. Identificar esta causa nos permitió ajustar la sensibilidad y ubicación del sensor, asegurando su desempeño adecuado en diferentes condiciones de iluminación.

7| ENLACES

A través del siguiente enlace se accede al repositorio GitHub con el trabajo completo.

<https://github.com/Soniacarrero/MICRO>