

**Carrera:** Ingeniería Electrónica

**Asignatura:** Técnicas Digitales II

**Año:** 2024

**Grupo N° 5**

**Listado de integrantes del grupo.**

**Alumnos:** Ammiraglia, Fabrizzio

Gutierrez, Abel

Sotomayor, Pablo

**Profesor:** Mansilla, Dario Ruben

## **Informe de proyecto Final**

**Título del Proyecto Final:**

### **1.- Consideraciones sobre el hardware del proyecto**

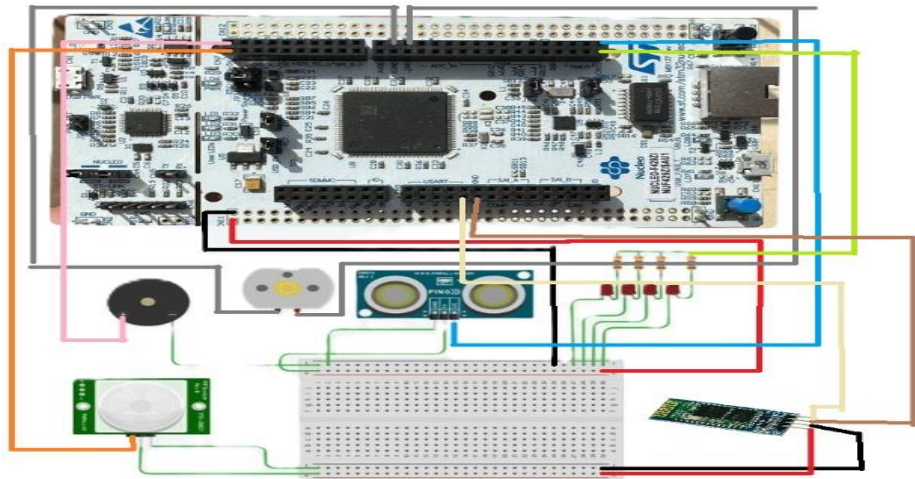
Los componentes por usar en el proyecto fueron considerados por la rentabilidad, practicidad y durabilidad que nos permitirán a lo largo del tiempo, siendo así un proyecto duradero en el tiempo con una gran precisión en cada aspecto que vaya a cumplir siendo así totalmente confiable para el usuario

#### **1.1. Descripción del proyecto.** Objetivos. Funcionamiento. Diagrama en Bloques.

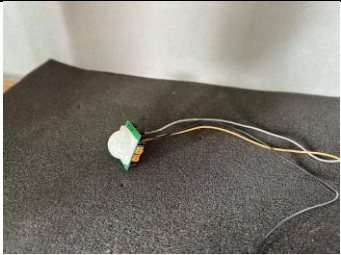
El objetivo de este proyecto es automatizar las actividades cotidianas de las personas para facilitar su largo día, brindado en este caso, seguridad y ahorro de energía.






El funcionamiento de este proyecto es de la siguiente manera siendo que por medio de sensores (puntualmente PIR y ultrasónico) manejar el uso de las luces en la sala de la cocina y sala de estar, y a su vez dando seguridad a través de una alarma. Y no solo eso, sino que también otro funcionamiento que tiene es la practicidad para el portón de la casa por medio de un módulo bluetooth.

#### **1.2. Circuito del proyecto.** Plano eléctrico, esquemático. Conexionado con la placa de desarrollo utilizada. Conexión con sensor/es, con display, con módulos externos utilizados. PCB de la placa externa que se haya implementado, si corresponde.



1.3. Listado de componentes.

Componentes	Cantidad	Imagen
Sensor PIR	1	

Sensor IR	1	
Modulo bluetooth HC-06	1	
LCD 16x2 c/ modulo I2C	1	
Placa nucleo f429zi	1	
Leds	2	
Buzzer	1	

--	--	--

## 2.- Consideraciones sobre el software

Lo que consideramos para el software fue los condicionales y las funciones no bloqueantes lo cual nos permitía el funcionamiento óptimo en todo momento del proyecto, minimizando el margen de que falle en algún momento algo

### 1.1. Link al repositorio

[https://github.com/Fabri09567/Grupo\\_5\\_TDII\\_2024](https://github.com/Fabri09567/Grupo_5_TDII_2024)

### LINK DEL VIDEO CON SU FUNCIONAMIENTO

<https://youtube.com/shorts/tXxtT-tGZQ?feature=share>

**Nombre del Proyecto:** *Proyecto*

### 1.2. Descripción de funcionamiento de la aplicación del proyecto.

La aplicación se basa en una placa de desarrollo STM32-Nucleo-F429ZI y utiliza un sensor IR que se encargara de detectar movimiento, en caso de haberlo, lo cual activara la alarma emitiendo un sonido a través de un buzzer.pero a su vez dando el tiempo necesario a ser desactivada por el usuario.

También utilizando un sensor PIR, el cual nos permitirá detectar movimiento, para así prender las luces de la sala común de la casa en caso de estar siendo ocupada en su momento. Y que con el pasar del tiempo, esto nos permita más allá de la automatización, el ahorro de la energía

Otro modulo que utilizamos y como antes mencionamos, fue un módulo bluetooth que nos permitirá a través de una app, manejar la activación de estos ambos sensores

### 1.3. Listado de los módulos de software desarrollados en el proyecto.

Los módulos que utilizamos para el desarrollo del software fueron:

- led\_on y led\_off: Manejara el encendido y apagado de los leds cuando el sensor pir se coloque en high
- Alarma\_desactivada y Alarma\_activada: Esta encargada de largar un mensaje en el lcd que dira el estado de la alarma
- Alarma\_conteo: Manejara el conteo del tiempo necesario para que el usuario pueda desactivar la alarma
- Alarma\_sonando: Nos avisara de que alarma se encuentra sonando en dicho momento

Mas allá de estas funciones, también tenemos las funciones básicas como los API que se desarrollan en el apartado siguiente

**Utilizo en la carpeta API los siguientes módulos:**

**API\_PIR:** Compuesto por toda la parte lógica necesaria para hacer funcionar el el sensor pir y su parte lógicas al momento de utilizar este

**API\_GPIO:** Está compuesto por la inicialización del módulo gpio como así también por funciones que manejan el encendido y apagado de los leds y leer sus respectivos valores

**API\_IR:** Compuesto por toda la parte lógica necesaria para hacer funcionar el el sensor IR y sus partes lógicas al momento de utilizar este

**API\_I2C\_LCD:** Este módulo contiene las funciones que manejan la comunicación con el LCD mediante I2C. Son de terceros, pero adaptadas a los parámetros de la placa STM utilizada.

**Nota:** En cada módulo API se ubica la función de inicialización y configuración del módulo correspondiente utilizado por este.

#### 1.4. Listado de los periféricos que utiliza en el proyecto.

Son los siguientes:

**GPIO:** Para manejo de los LEDs de usuario y la comunicación con el sensor ultrasónico.

**I2C1:** Para manejo de la comunicación con el LCD.

**USART3:** Es el encargado del manejo del modulo bluetooth

#### Datasheet de los modulos de hardware

##### Sensor IR

<https://agelectronica.lat/pdfs/textos/O/OKY3127.PDF>

##### Sensor pir

<https://www.alldatasheet.es/datasheet-pdf/download/1131987/ETC2/HCSR501.html>

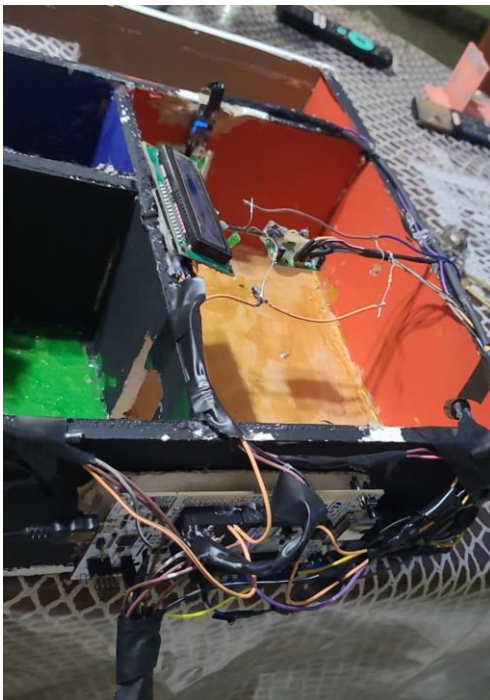
##### Sensor bluetooth

<https://www.alldatasheet.es/datasheet-pdf/pdf/1179032/ETC1/HC-06.html>

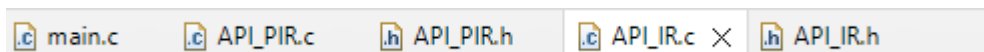
##### Placa stm32 nucleo f429zi

<https://www.st.com/en/evaluation-tools/nucleo-f429zi.html>

#### FOTOS DE LA MAQUETA DEL PROYECTO



## Capturas de la programacion





```

main.c API_PIR.c API_PIR.h API_IR.c x API_IR.h
9  #include "i2c.h"
10 #include "usart.h"
11 #include "gpio.h"
12 #include "API_liquidcrystal_i2c.h"
13 #include "stm32f4xx_hal.h"
14 #include "string.h"
15
16 #define SENSOR_IR_PIN GPIO_PIN_0 // Pin de entrada del sensor IR (PA0)
17 #define BUZZER_PIN GPIO_PIN_0 // Pin de salida del buzzer (PB0)
18
19 extern uint32_t tiempoDeteccionIR;
20 extern uint8_t modoIR;
21 extern uint8_t zumbadorActivadoIR;
22 extern uint8_t conteoIRIniciado;
23
24 void activando_alarma(void){
25     HD44780_Clear();
26     HD44780_SetCursor(0, 0);
27     HD44780_PrintStr("Activando alarma");
28     HAL_Delay(3000);
29     HD44780_Clear();
30     HD44780_SetCursor(0, 0);
31     HD44780_PrintStr("Alarma activada");
32     HAL_Delay(100);
33 }
34
35 void alarma_sonando(void){
36     HAL_GPIO_WritePin(GPIOB, BUZZER_PIN, GPIO_PIN_SET); //encender el zumbador.
37     HD44780_Clear();
38     HD44780_SetCursor(0, 0);
39     HD44780_PrintStr("Alarma Sonando");
40     HAL_Delay(100);
41 }
42
43 void alarma_conteo(void){
44     if (HAL_GetTick() - tiempoDeteccionIR >= 3000) {
45         // 3 segundos transcurridos, verificar si se cambi6 a modo PIR
46         if (modoIR) {
47             zumbadorActivadoIR = 1;
48
49 while (1)
50 {
51     /* USER CODE END WHILE */
52     if (modoIR) {
53
54         HAL_GPIO_WritePin(GPIOD, LED_PIN, GPIO_PIN_RESET);
55         if (alarmaIRActivada == 0) {
56             activando_alarma();
57             // Esperar 3 segundos antes de activar el sensor IR
58             if (tiempoActivacionIR == 0) {
59                 tiempoActivacionIR = HAL_GetTick();
60             }
61
62             if (HAL_GetTick() - tiempoActivacionIR >= 3000) {
63                 alarmaIRActivada = 1; // Activar alarma IR después de 3 segundos
64                 tiempoActivacionIR = 0; // Reiniciar el tiempo de activación
65             }
66         } else {
67             // Modo IR activado
68             GPIO_PinState sensorEstadoIR = HAL_GPIO_ReadPin(GPIOA, SENSOR_IR_PIN);
69
70             if (sensorEstadoIR == GPIO_PIN_RESET && conteoIRIniciado == 0) {
71                 // Detección de movimiento IR, iniciar conteo
72                 tiempoDeteccionIR = HAL_GetTick();
73                 conteoIRIniciado = 1;
74             }
75
76             if (conteoIRIniciado) {
77                 alarma_conteo();
78             }
79             if (zumbadorActivadoIR == 1){
80                 alarma_sonando();
81             }else{
82                 HAL_GPIO_WritePin(GPIOB, BUZZER_PIN, GPIO_PIN_RESET);
83             }
84         }
85     } else {
86

```