UTN-FRT
TÉCNICAS DIGITALES II
ING. ELECTRÓNICA

SISTEMA DE SEGURIDAD
ALARMA CON COMUNICACIÓN BLUETOOTH

GRUPO 8

Alumnos: Gao, Luciano

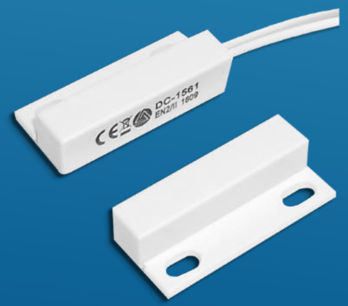
Jorrat, Tomás

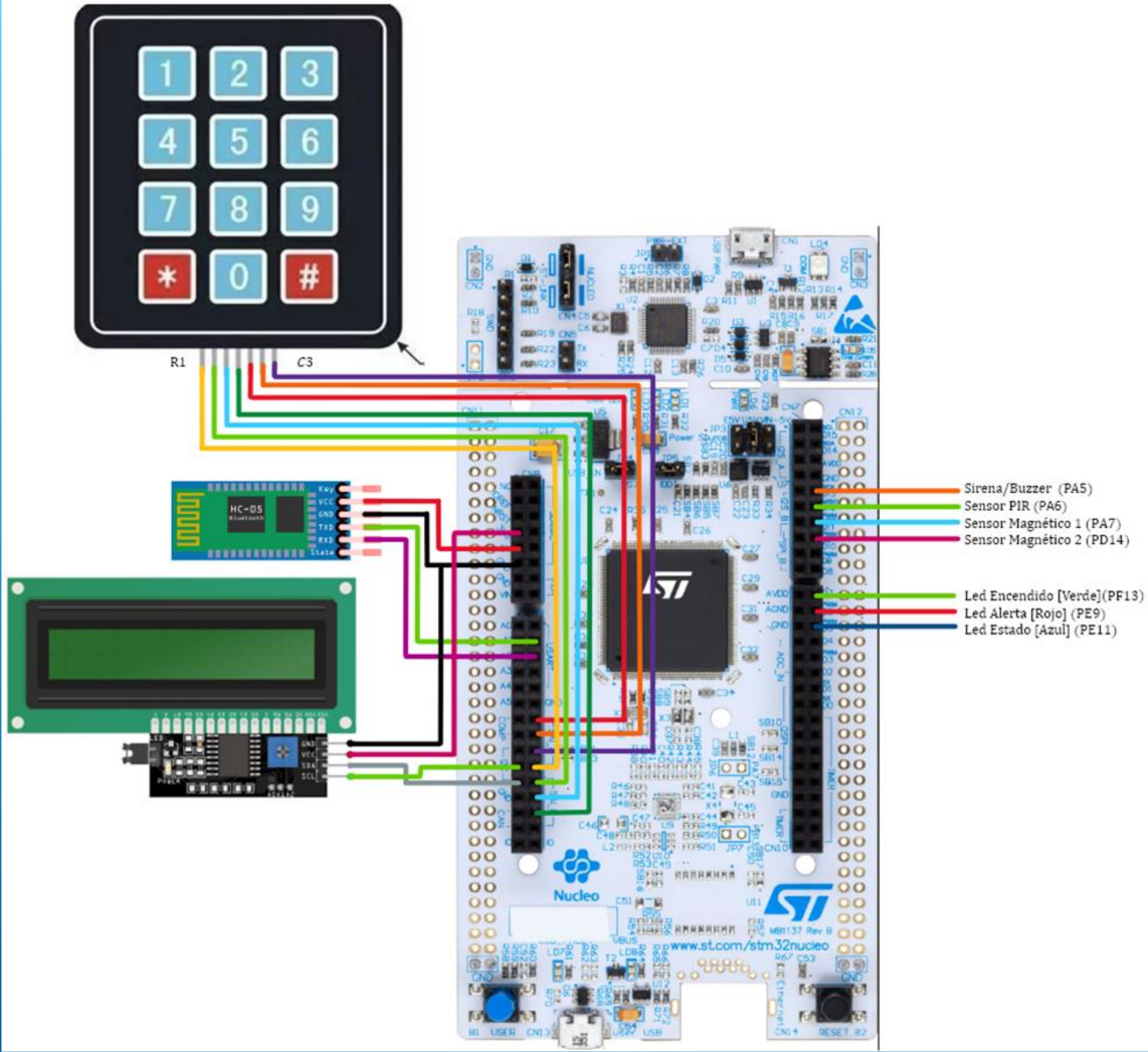
Mitre, Emilio

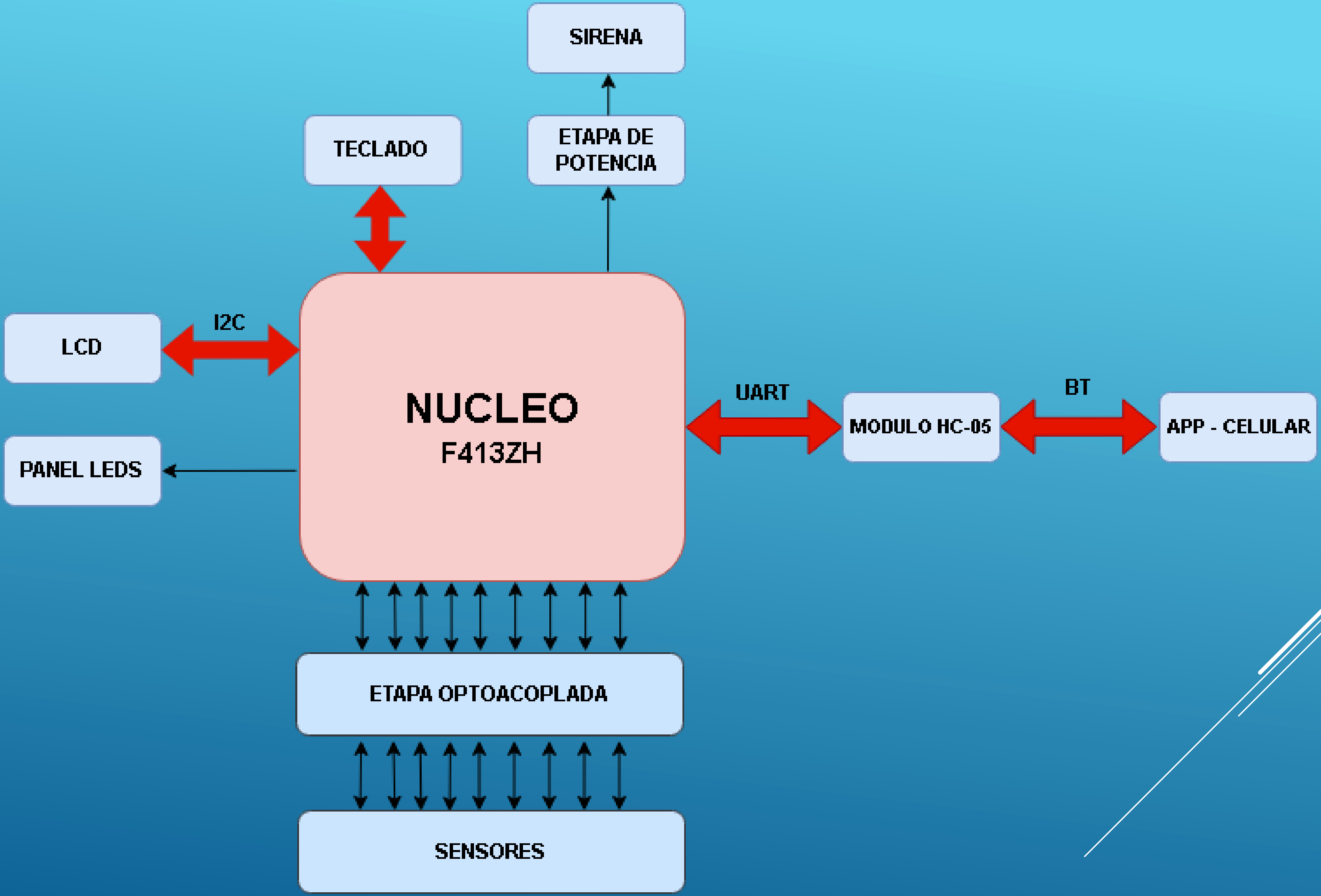
Profesor: Ing. Mansilla Ruben Dario

STM32-F413ZH

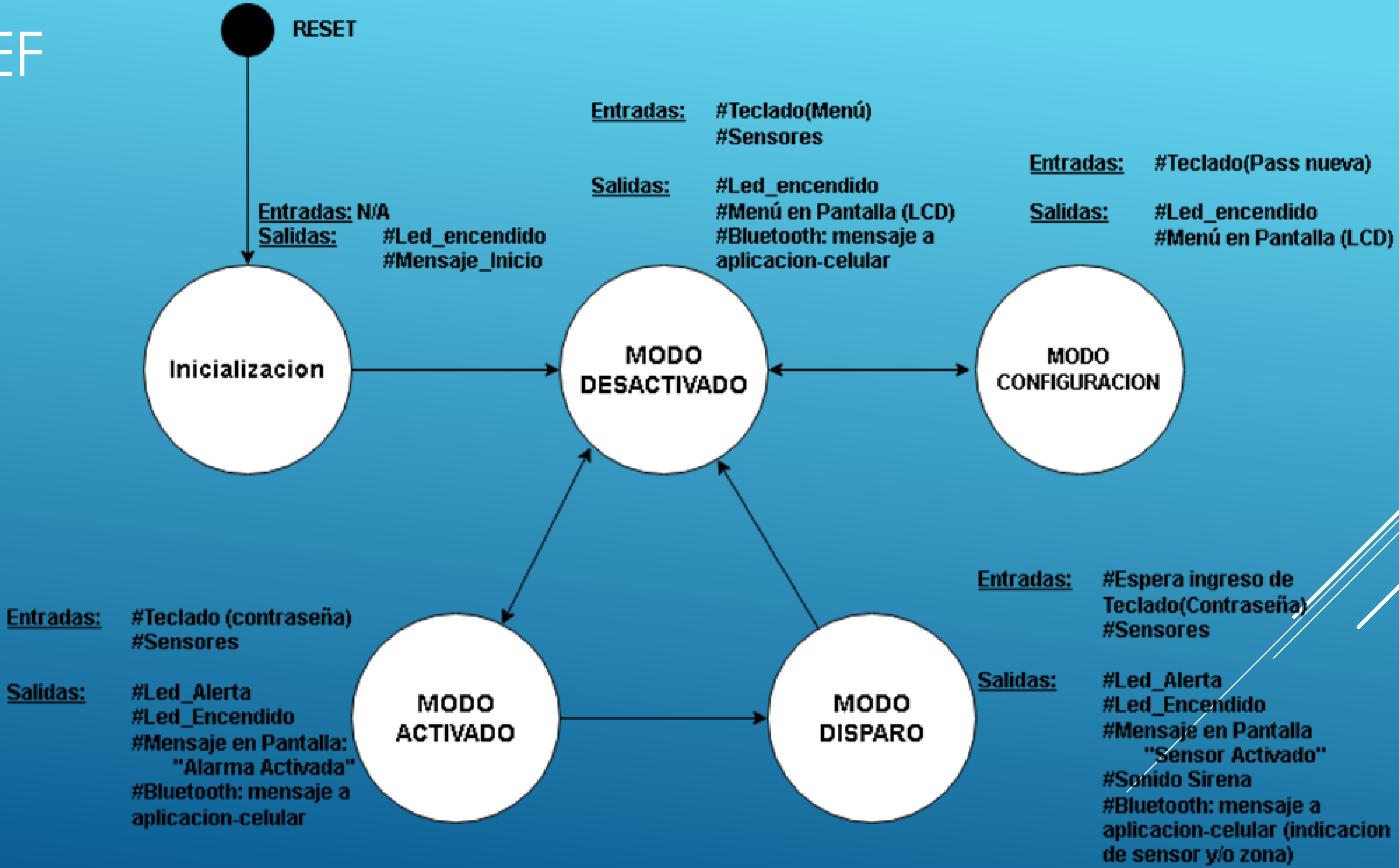
Teclado Matricial 4x3
Modulo Bluetooth HC-05
Pantalla LCD 16x2 - PCF8574
Sensor PIR **HC-SR501**
Sensor Magnético MS-14-BL

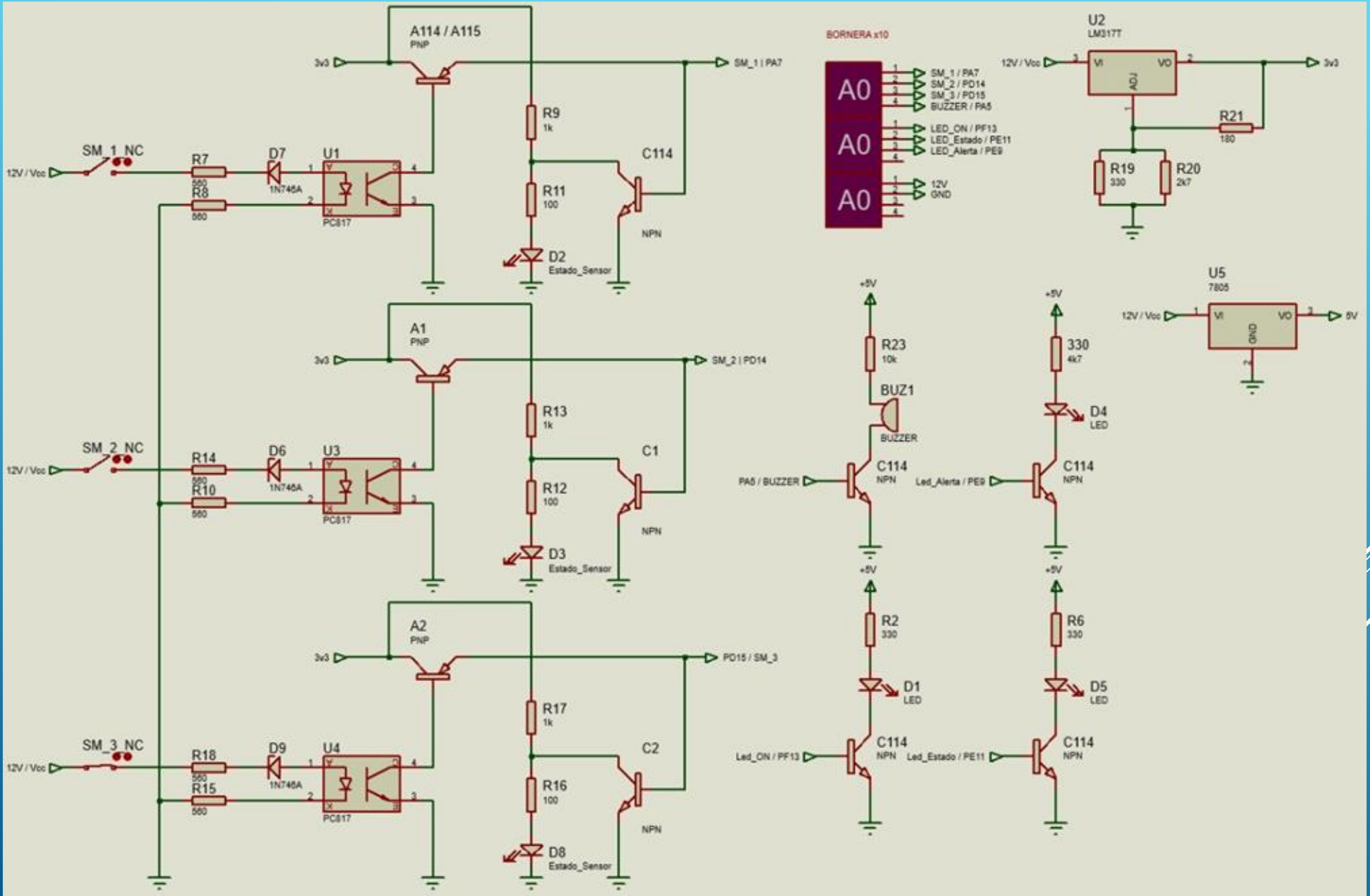


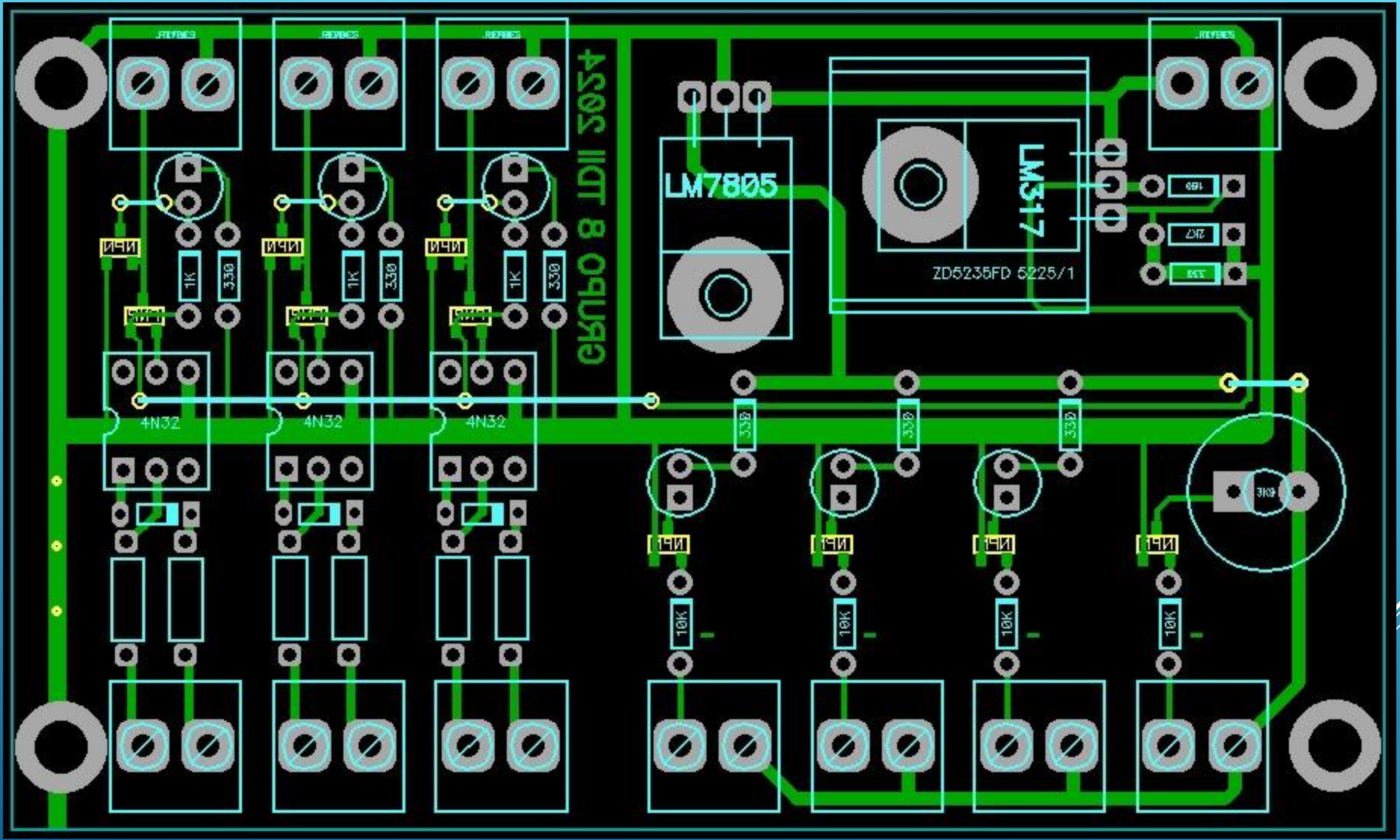



SIRENA
TECLADO
ETAPA DE POTENCIA
LCD
I2C
NUCLEO
F413ZH
UART
MODULO HC-05
BT
APP - CELULAR
PANEL LEDS
ETAPA OPTOACOPLADA
SENSORES

MEF

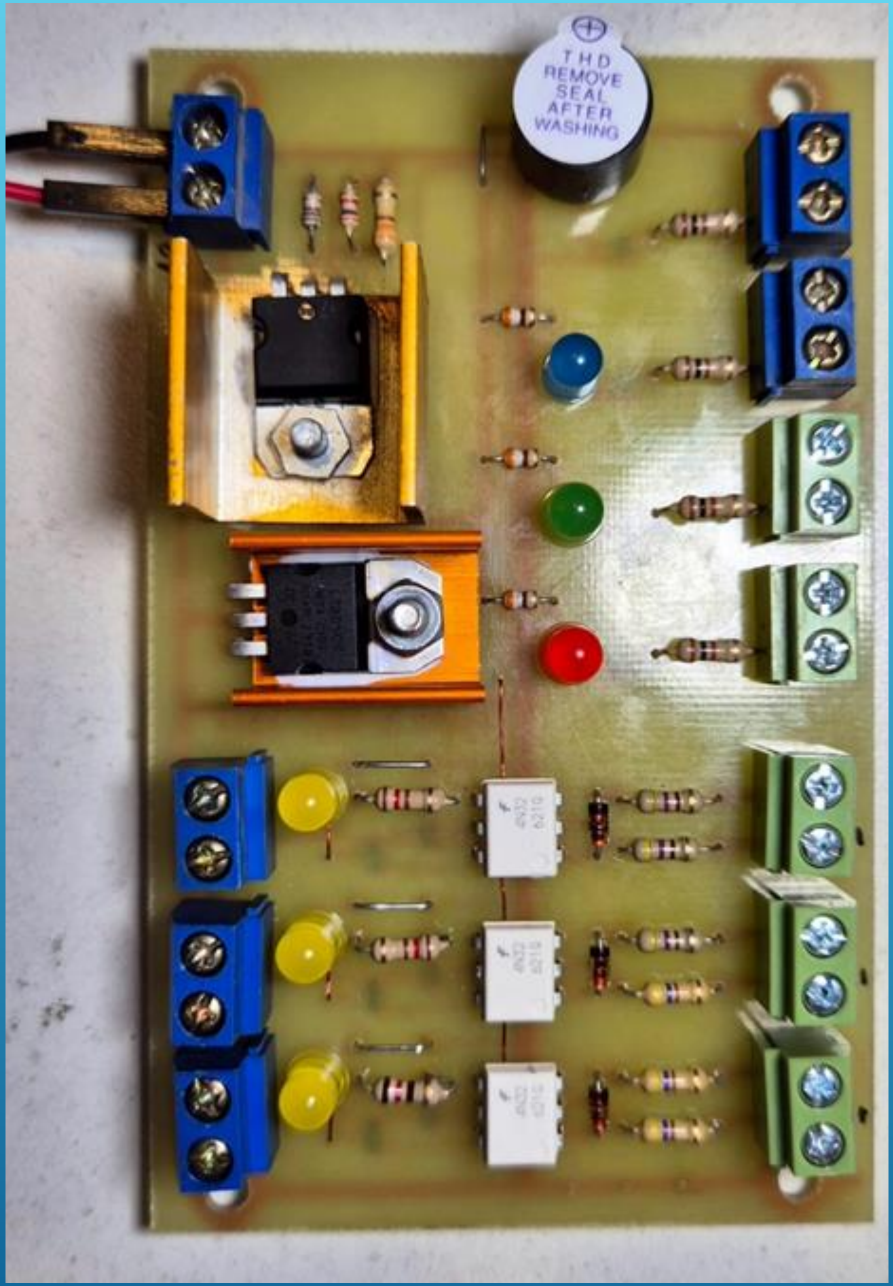

RESET
Entradas: N/A
Salidas: #Led_encendido
#Mensaje_Inicio
Inicializacion
Entradas: #Teclado(Menú)
#Sensores
Salidas: #Led_encendido
#Menú en Pantalla (LCD)
#Bluetooth: mensaje a aplicacion-celular
MODO DESACTIVADO
Entradas: #Teclado(Pass nueva)
Salidas: #Led_encendido
#Menú en Pantalla (LCD)
MODO CONFIGURACION
Entradas: #Teclado (contraseña)
#Sensores
Salidas: #Led_Alerta
#Led_Encendido
#Mensaje en Pantalla: "Alarma Activada"
#Bluetooth: mensaje a aplicacion-celular
MODO ACTIVADO
MODO DISPARO
Entradas: #Espera ingreso de Teclado(Contraseña)
#Sensores
Salidas: #Led_Alerta
#Led_Encendido
#Mensaje en Pantalla "Sensor Activado"
#Sonido Sirena
#Bluetooth: mensaje a aplicacion-celular (indicacion de sensor y/o zona)

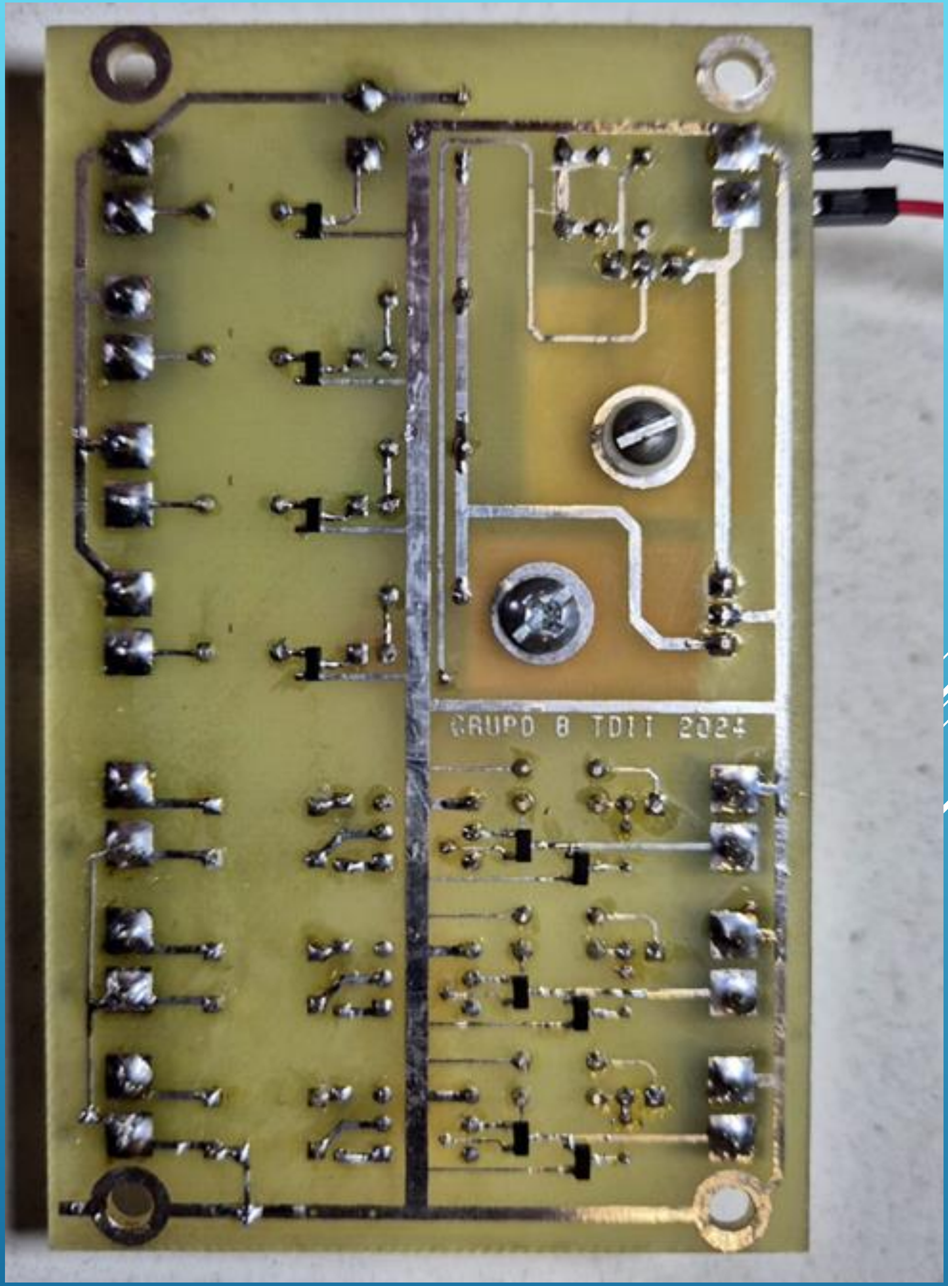

A114 / A115
PNP
3v3
SM_1 / PA7
R9
1k
C114
SM_1_NC
12V / Vcc
R7
560
D7
1N746A
U1
R8
560
PC817
R11
100
NPN
D2
Estado_Sensor
A1
PNP
3v3
SM_2 / PD14
R13
1k
C1
SM_2_NC
12V / Vcc
R14
560
D6
1N746A
U3
R10
560
PC817
R12
100
NPN
D3
Estado_Sensor
A2
PNP
3v3
PD15 / SM_3
R17
1k
C2
SM_3_NC
12V / Vcc
R18
560
D9
1N746A
U4
R15
560
PC817
R16
100
NPN
D8
Estado_Sensor
BORNERA x10
A0
A0
A0
SM_1 / PA7
SM_2 / PD14
SM_3 / PD15
BUZZER / PA5
LED_ON / PF13
LED_Estado / PE11
LED_Alerta / PE9
12V
GND
U2
LM317T
12V / Vcc
VI
VO
ADJ
3v3
R21
180
R19
330
R20
2k7
U5
7805
12V / Vcc
VI
VO
GND
5V
+5V
R23
10k
BUZ1
BUZZER
PA5 / BUZZER
C114
NPN
+5V
330
4k7
D4
LED
Led_Alerta / PE9
C114
NPN
+5V
R2
330
D1
LED
Led_ON / PF13
C114
NPN
+5V
R6
330
D5
LED
Led_Estado / PE11
C114
NPN


GRUPO 8 TDII 2024
LM7805
LM317
ZD5235FD 5225/1
4N32
4N32
4N32
1K
330
1K
330
1K
330
330
330
330
10K
10K
10K
10K
3K9

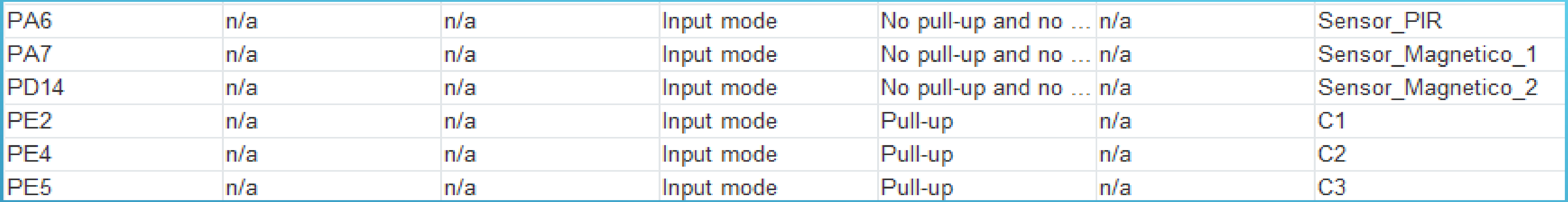
PCB:

Sensores Magneticos
Sensor PIR
Buzzer
Leds de estado

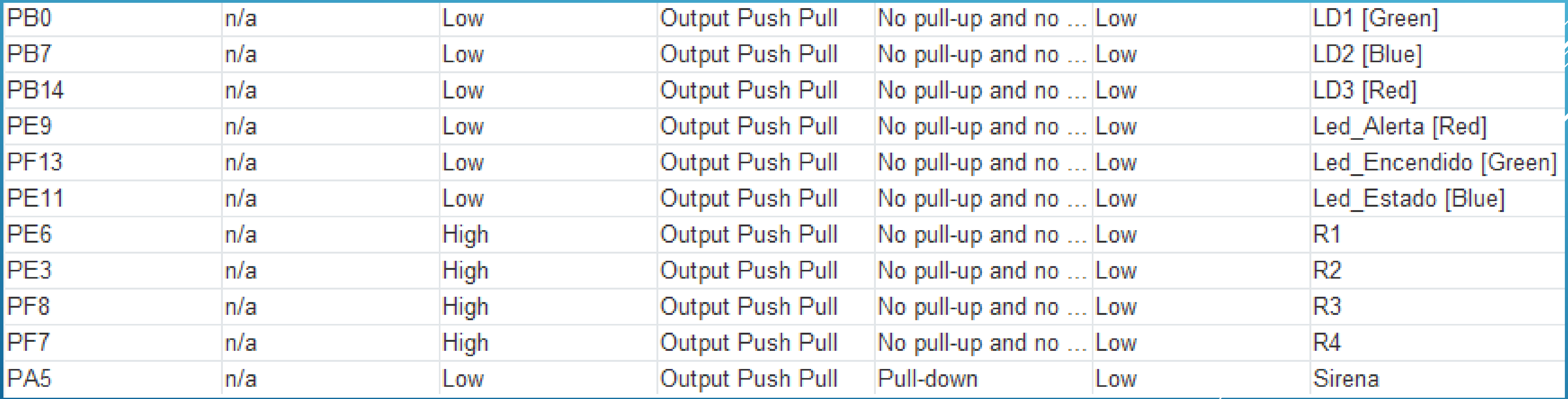




CONFIGURACIÓN DE PINES I/O

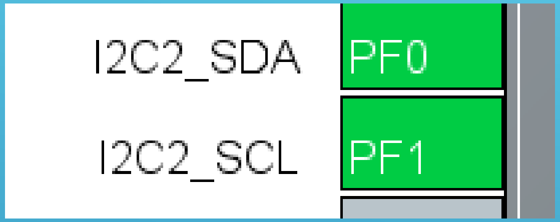


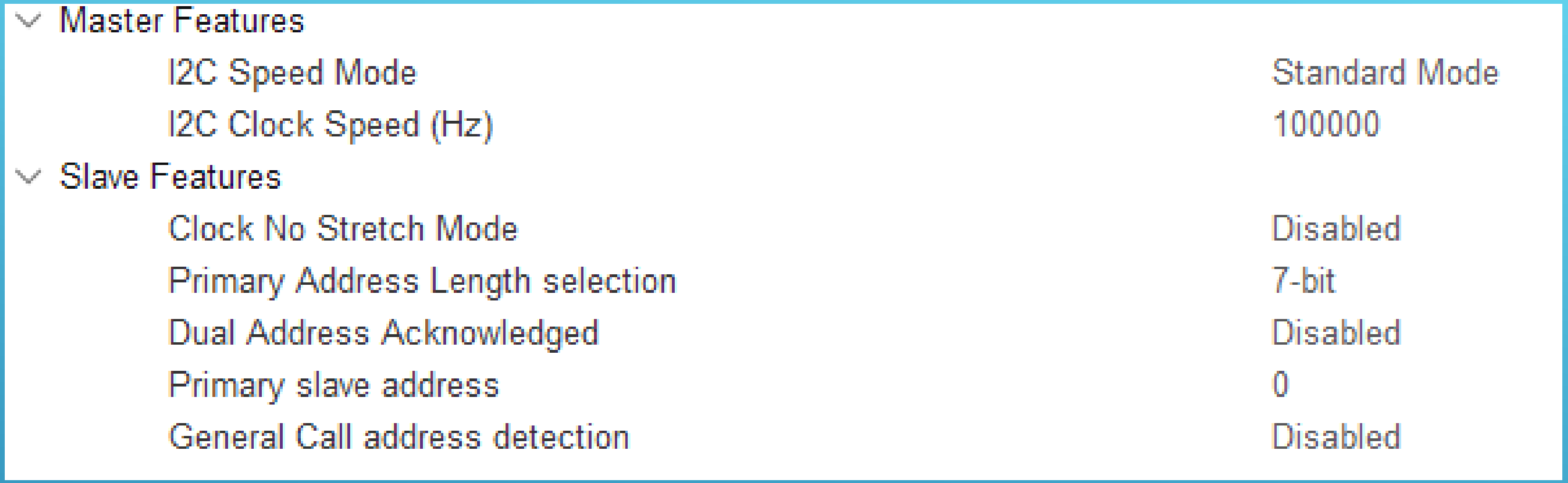
PA6	n/a	n/a	Input mode	No pull-up and no ...	n/a	Sensor_PIR
PA7	n/a	n/a	Input mode	No pull-up and no ...	n/a	Sensor_Magnetico_1
PD14	n/a	n/a	Input mode	No pull-up and no ...	n/a	Sensor_Magnetico_2
PE2	n/a	n/a	Input mode	Pull-up	n/a	C1
PE4	n/a	n/a	Input mode	Pull-up	n/a	C2
PE5	n/a	n/a	Input mode	Pull-up	n/a	C3



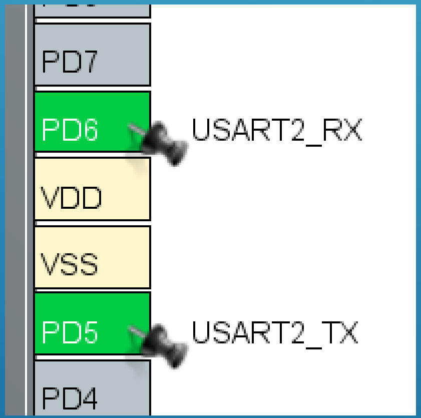
PB0	n/a	Low	Output Push Pull	No pull-up and no ...	Low	LD1 [Green]
PB7	n/a	Low	Output Push Pull	No pull-up and no ...	Low	LD2 [Blue]
PB14	n/a	Low	Output Push Pull	No pull-up and no ...	Low	LD3 [Red]
PE9	n/a	Low	Output Push Pull	No pull-up and no ...	Low	Led_Alerta [Red]
PF13	n/a	Low	Output Push Pull	No pull-up and no ...	Low	Led_Encendido [Green]
PE11	n/a	Low	Output Push Pull	No pull-up and no ...	Low	Led_Estado [Blue]
PE6	n/a	High	Output Push Pull	No pull-up and no ...	Low	R1
PE3	n/a	High	Output Push Pull	No pull-up and no ...	Low	R2
PF8	n/a	High	Output Push Pull	No pull-up and no ...	Low	R3
PF7	n/a	High	Output Push Pull	No pull-up and no ...	Low	R4
PA5	n/a	Low	Output Push Pull	Pull-down	Low	Sirena

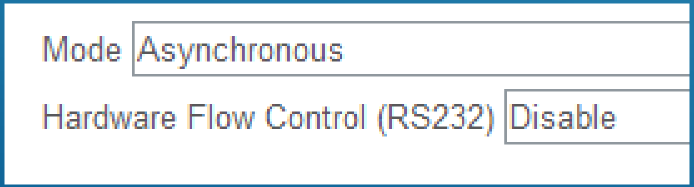
CONFIGURACIÓN DE MÓDULOS – USART2 - I2C2



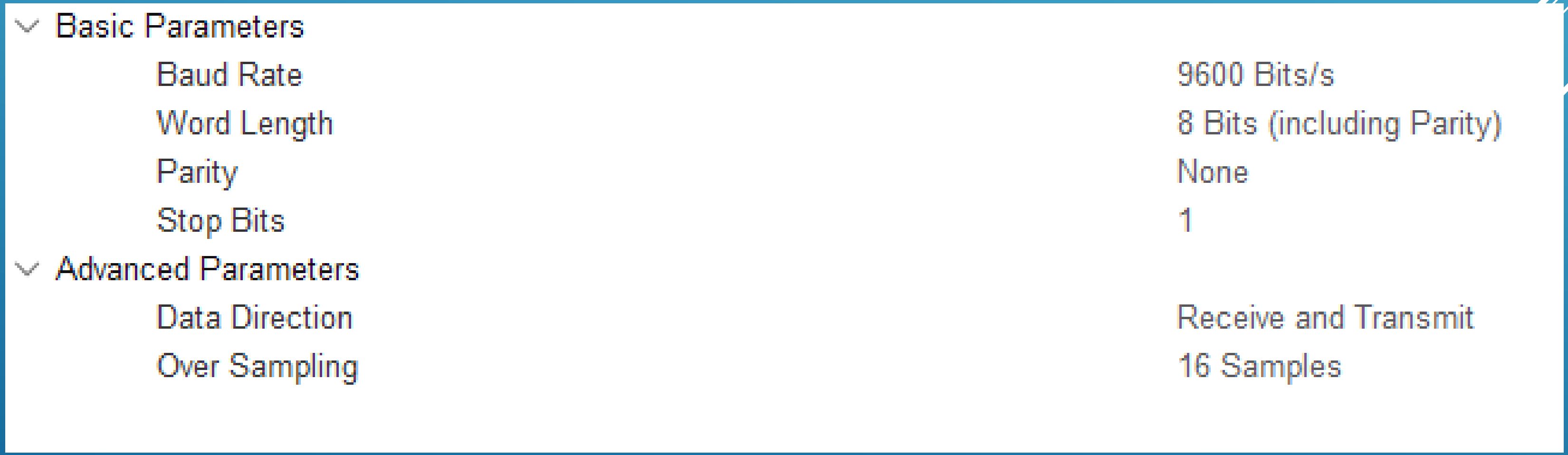


Master Features	
I2C Speed Mode	Standard Mode
I2C Clock Speed (Hz)	100000
Slave Features	
Clock No Stretch Mode	Disabled
Primary Address Length selection	7-bit
Dual Address Acknowledged	Disabled
Primary slave address	0
General Call address detection	Disabled



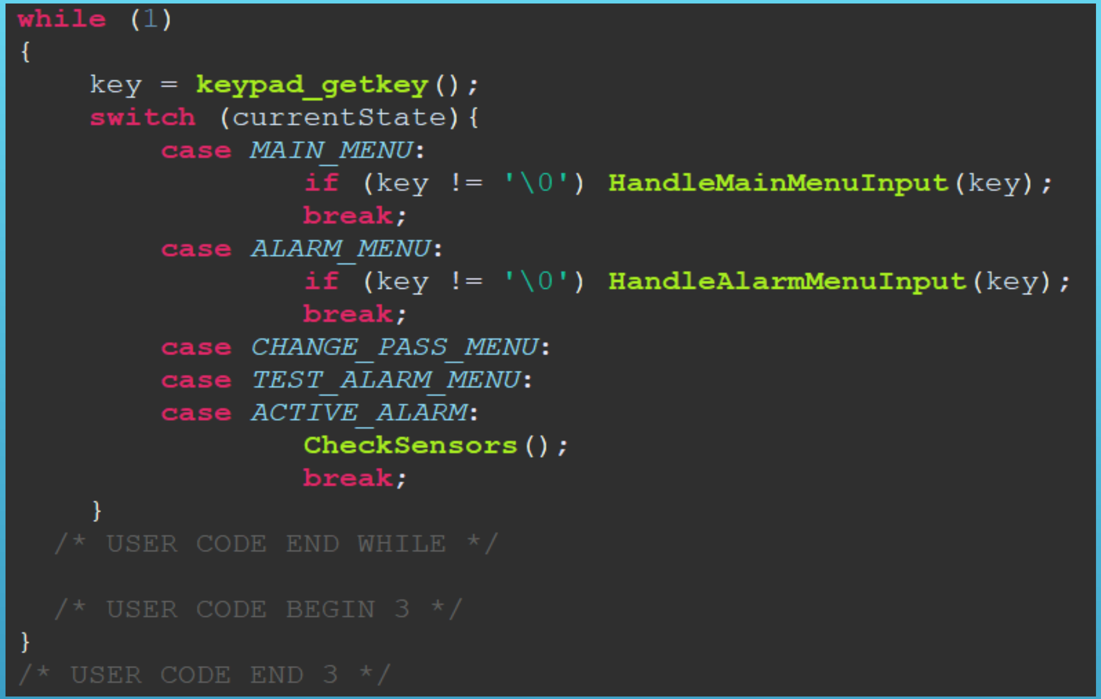


Mode	Asynchronous
Hardware Flow Control (RS232)	Disable

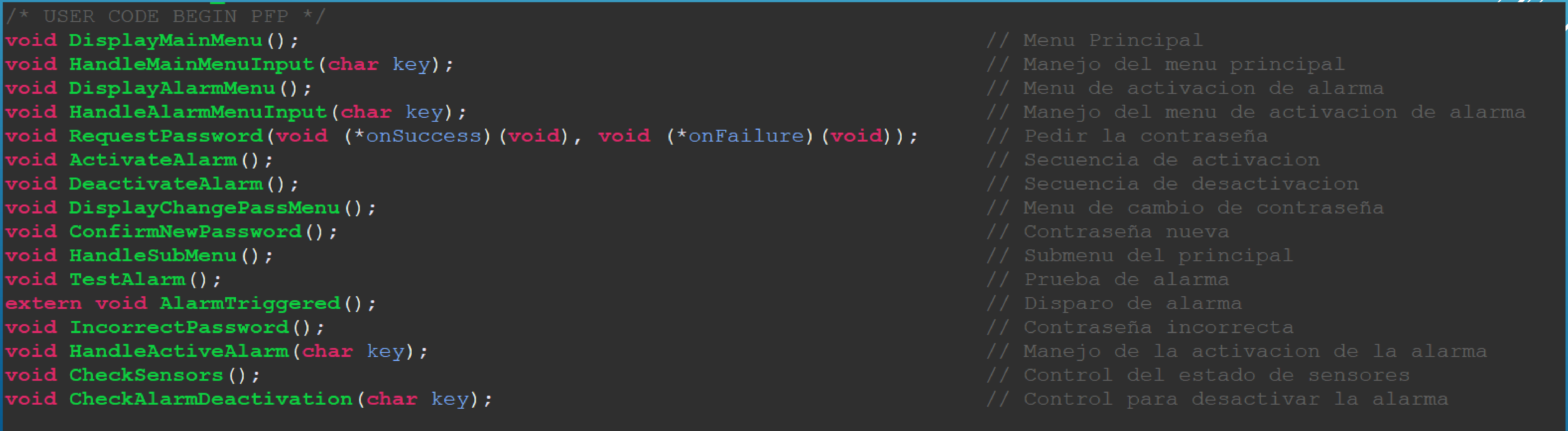


Basic Parameters	
Baud Rate	9600 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
Advanced Parameters	
Data Direction	Receive and Transmit
Over Sampling	16 Samples

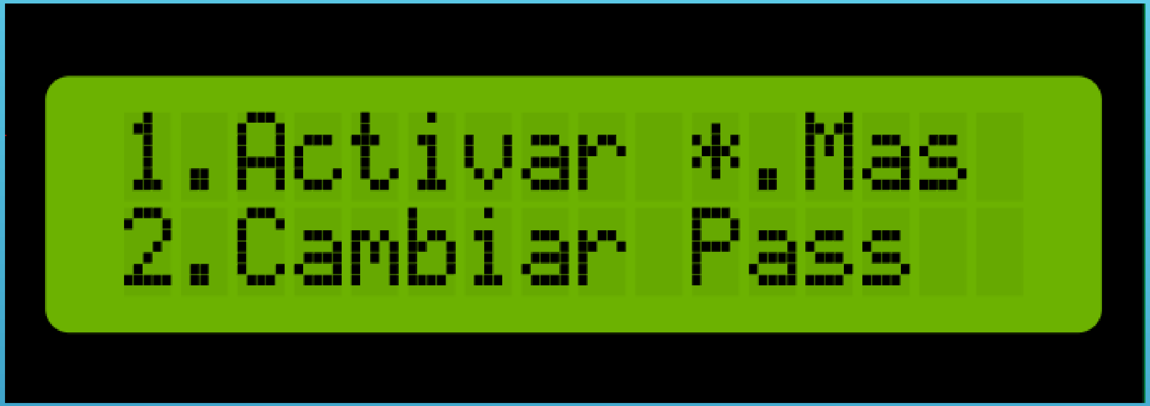
SOFTWARE – MAIN.C



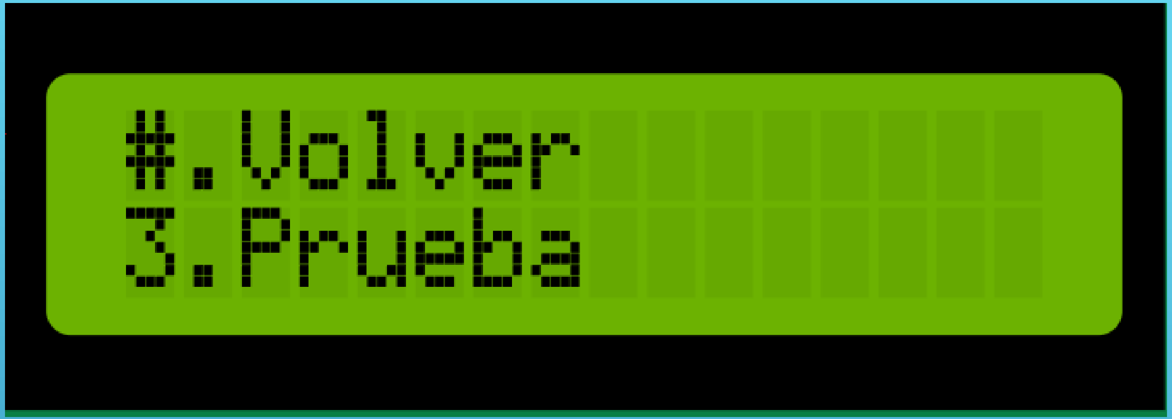
```c
while (1)
{
    key = keypad_getkey();
    switch (currentState){
        case MAIN_MENU:
                if (key != '\0') HandleMainMenuInput(key);
                break;
        case ALARM_MENU:
                if (key != '\0') HandleAlarmMenuInput(key);
                break;
        case CHANGE_PASS_MENU:
        case TEST_ALARM_MENU:
        case ACTIVE_ALARM:
                CheckSensors();
                break;
    }
  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
```



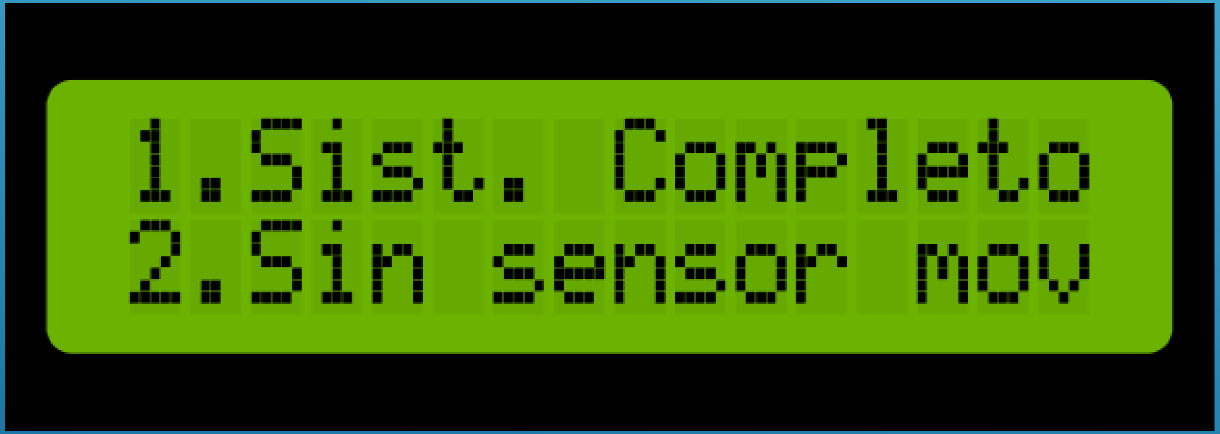
```c
/* USER CODE BEGIN PFP */
void DisplayMainMenu();                                          // Menu Principal
void HandleMainMenuInput(char key);                              // Manejo del menu principal
void DisplayAlarmMenu();                                         // Menu de activacion de alarma
void HandleAlarmMenuInput(char key);                             // Manejo del menu de activacion de alarma
void RequestPassword(void (*onSuccess)(void), void (*onFailure)(void)); // Pedir la contraseña
void ActivateAlarm();                                            // Secuencia de activacion
void DeactivateAlarm();                                          // Secuencia de desactivacion
void DisplayChangePassMenu();                                    // Menu de cambio de contraseña
void ConfirmNewPassword();                                       // Contraseña nueva
void HandleSubMenu();                                            // Submenu del principal
void TestAlarm();                                                // Prueba de alarma
extern void AlarmTriggered();                                    // Disparo de alarma
void IncorrectPassword();                                        // Contraseña incorrecta
void HandleActiveAlarm(char key);                                // Manejo de la activacion de la alarma
void CheckSensors();                                             // Control del estado de sensores
void CheckAlarmDeactivation(char key);                           // Control para desactivar la alarma
```



```
1.Activar *.Mas
2.Cambiar Pass
```

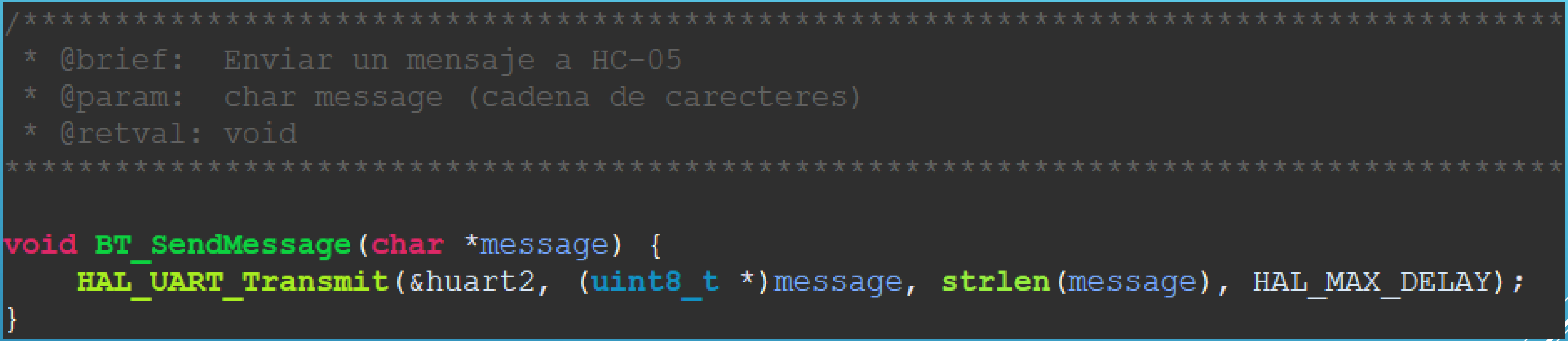


```
#.Volver
3.Prueba
```

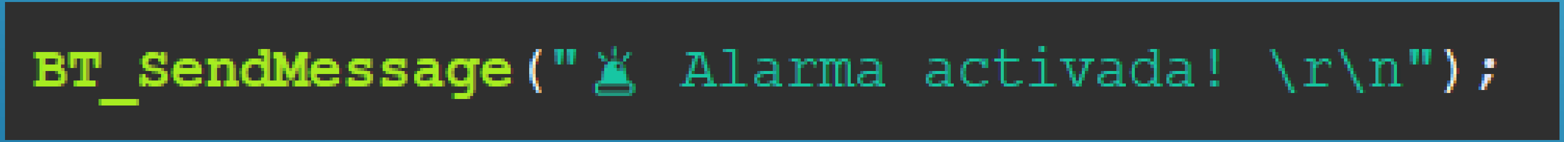


```
1.Sist. Completo
2.Sin sensor mov
```

API_BT

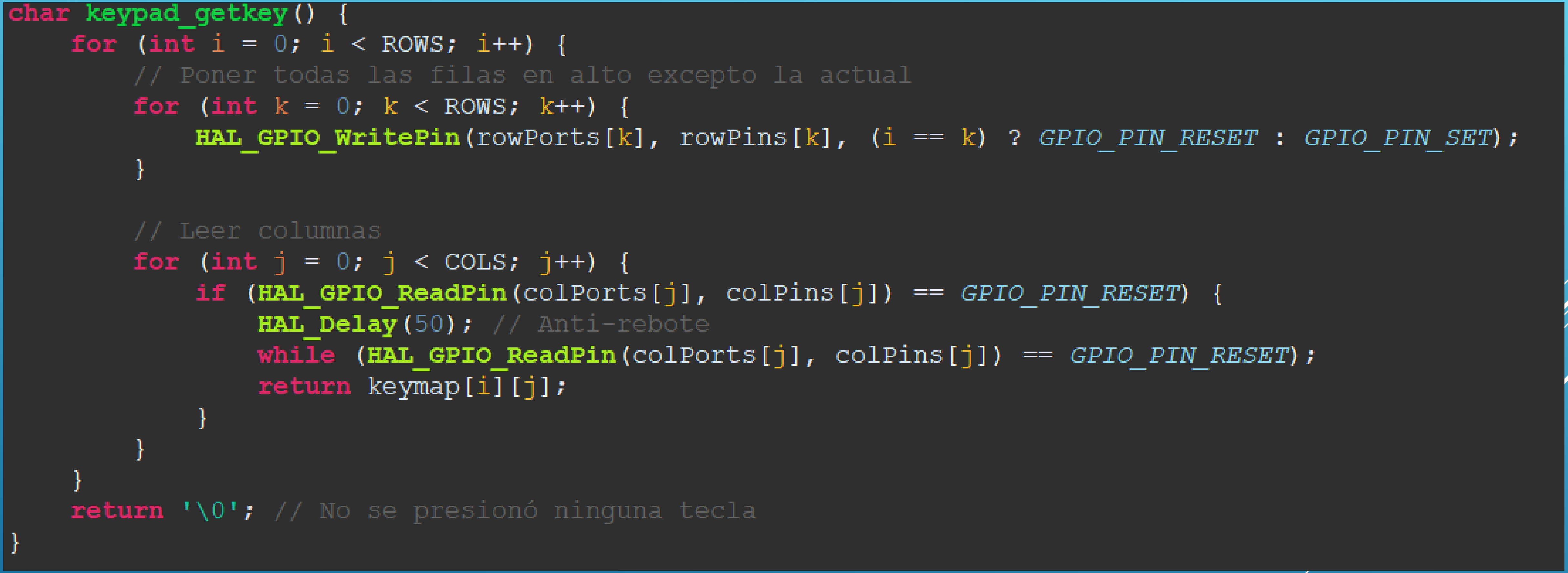


```c
/************************************************************************************
 * @brief:  Enviar un mensaje a HC-05
 * @param:  char message (cadena de carecteres)
 * @retval: void
*************************************************************************************

void BT_SendMessage(char *message) {
    HAL_UART_Transmit(&huart2, (uint8_t *)message, strlen(message), HAL_MAX_DELAY);
}
```

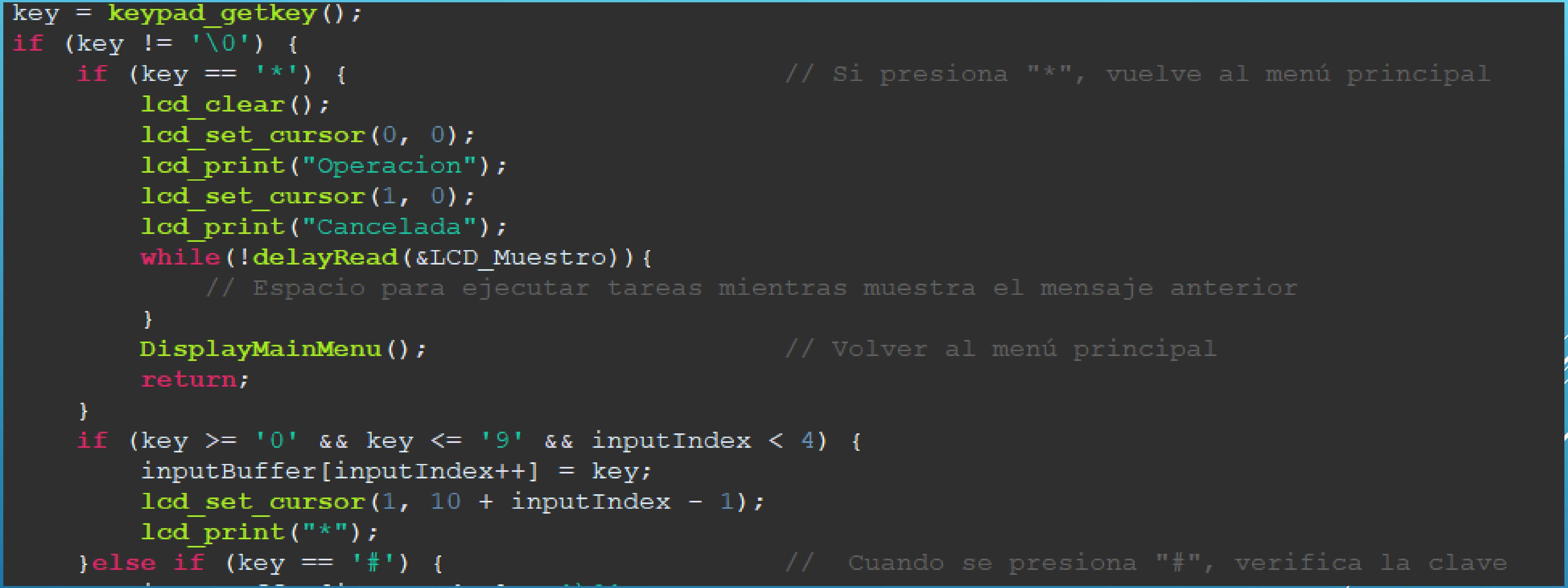


```c
BT_SendMessage("🚨 Alarma activada! \r\n");
```

API_KEYPAD



```c
char keypad_getkey() {
    for (int i = 0; i < ROWS; i++) {
        // Poner todas las filas en alto excepto la actual
        for (int k = 0; k < ROWS; k++) {
            HAL_GPIO_WritePin(rowPorts[k], rowPins[k], (i == k) ? GPIO_PIN_RESET : GPIO_PIN_SET);
        }

        // Leer columnas
        for (int j = 0; j < COLS; j++) {
            if (HAL_GPIO_ReadPin(colPorts[j], colPins[j]) == GPIO_PIN_RESET) {
                HAL_Delay(50); // Anti-rebote
                while (HAL_GPIO_ReadPin(colPorts[j], colPins[j]) == GPIO_PIN_RESET);
                return keymap[i][j];
            }
        }
    }
    return '\0'; // No se presionó ninguna tecla
}
```


```
key = keypad_getkey();
if (key != '\0') {
    if (key == '*') {                                    // Si presiona "*", vuelve al menú principal
        lcd_clear();
        lcd_set_cursor(0, 0);
        lcd_print("Operacion");
        lcd_set_cursor(1, 0);
        lcd_print("Cancelada");
        while(!delayRead(&LCD_Muestro)){
            // Espacio para ejecutar tareas mientras muestra el mensaje anterior
        }
        DisplayMainMenu();                               // Volver al menú principal
        return;
    }
    if (key >= '0' && key <= '9' && inputIndex < 4) {
        inputBuffer[inputIndex++] = key;
        lcd_set_cursor(1, 10 + inputIndex - 1);
        lcd_print("*");
    }else if (key == '#') {                              //  Cuando se presiona "#", verifica la clave
```

FIN

- Repositorio GRUPO 8: https://github.com/TomasJorrat/AFP_5_GRUPO_8_2024

- https://github.com/emiliomitre/Grupo_8_TDII_2024