

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC CẦN THƠ  
TRƯỜNG BÁCH KHOA  
KHOA ĐIỆN TỬ - VIỆN THÔNG  
NGÀNH KỸ THUẬT MÁY TÍNH

-----❖-----



**BÁO CÁO ĐỒ ÁN  
KỸ THUẬT VI XỬ LÝ**

**HỆ THỐNG GIÁM SÁT MÔI TRƯỜNG  
TRONG NHÀ KÍNH**

**CÁN BỘ HƯỚNG DẪN**  
**ThS. Trần Hữu Danh**

**NHÓM SV THỰC HIỆN:**

**Nguyễn Hoàng Khang**

**MSSV: B2308300**

**Lê Tiến Đạt**

**MSSV: B2308281**

**Ngành: Kỹ thuật máy tính – Khóa: 49**

**Tháng 08/2025**



---

# MỤC LỤC

	<i>Trang</i>
<i>MỤC LỤC</i> .....	<i>i</i>
<b>CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI</b> .....	<b>1</b>
1.1. Mục tiêu .....	1
1.2. Phạm vi và giới hạn .....	1
1.3. Phương pháp thực hiện.....	1
<b>CHƯƠNG II: NỘI DUNG VÀ KẾT QUẢ</b> .....	<b>2</b>
2.1. Thiết kế hệ thống.....	2
2.2. Kết quả đạt được.....	10
<b>CHƯƠNG III: KẾT LUẬN</b> .....	<b>12</b>
<b>TÀI LIỆU THAM KHẢO</b> .....	<b>12</b>



---

# CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

## 1.1. Mục tiêu

### **Giám sát các thông số môi trường trong nhà kính:**

Đo và hiển thị nhiệt độ, độ ẩm, chất lượng không khí, và cường độ ánh sáng để đảm bảo điều kiện môi trường phù hợp cho cây trồng.

### **Cảnh báo khi môi trường vượt ngưỡng:**

Sử dụng LED đơn để cảnh báo khi chất lượng không khí ở mức tệ hoặc nhiệt độ vượt mức.

### **Hiển thị thông tin rõ ràng, dễ quan sát:**

LCD 16x2 để hiển thị chất lượng không khí và nhiệt độ. LED 7 đoạn đôi để hiển thị độ ẩm. LED matrix 8x8 để hiển thị biểu đồ ánh sáng.

### **Tạo hệ giám sát môi trường với chi phí thấp:**

Dựa trên vi điều khiển STM32F411CEU6, sử dụng cảm biến thông dụng và nút nhấn vật lý, không cần kết nối không dây từ đó giảm chi phí đáng kể.

## 1.2. Phạm vi và giới hạn

### **Phạm vi:**

Hệ thống đo lường đơn giản, giao tiếp qua màn hình LCD và nút ấn vật lý.

### **Giới hạn:**

Không có kết nối không dây tới các thiết bị ngoài, màn hình cảm ứng, điều khiển qua giọng nói, cử chỉ,...

## 1.3. Phương pháp thực hiện

### **Phần cứng:**

- + Vi điều khiển STM32 để điều khiển.
- + Sử dụng LCD 16x2 để giao tiếp qua nút nhấn đơn giản, hiển thị thông tin.
- + LED ma trận 8x8 hiển thị biểu đồ cường độ ánh sáng.
- + Sử dụng quang trở để đo cường độ ánh sáng và cảm biến chất lượng không khí để đo chất lượng không khí.
- + Dùng cảm biến đo nhiệt độ và độ ẩm.
- + LED đơn dùng để cảnh báo khi nhiệt độ vượt mức giới hạn hoặc chất lượng không khí ở mức tệ.
- + Nút nhấn dùng để điều khiển.

### **Phần mềm:**

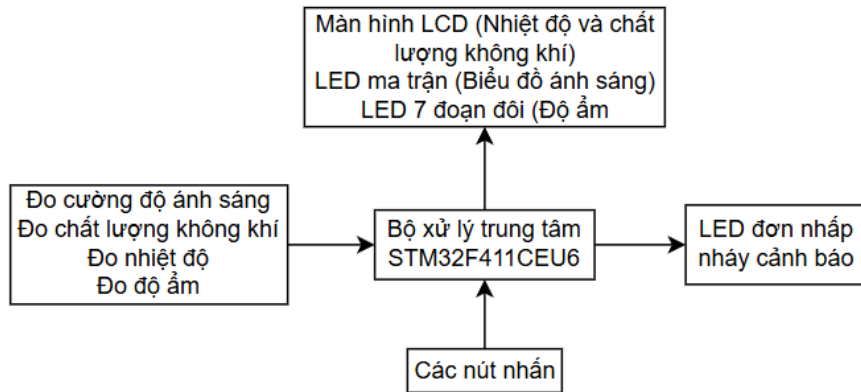
- + Sử dụng STM32CubeIDE để lập trình C theo chuẩn HAL.
- + Sử dụng các chân GPIO, ADC để đọc giá trị từ các cảm biến.
- + Sử dụng các chân GPIO để điều khiển hoạt động của LCD, LED ma trận, LED 7 đoạn đôi, LED đơn.



## CHƯƠNG II: NỘI DUNG VÀ KẾT QUẢ

### 2.1. Thiết kế hệ thống

Sơ đồ khối tổng quát của hệ thống:

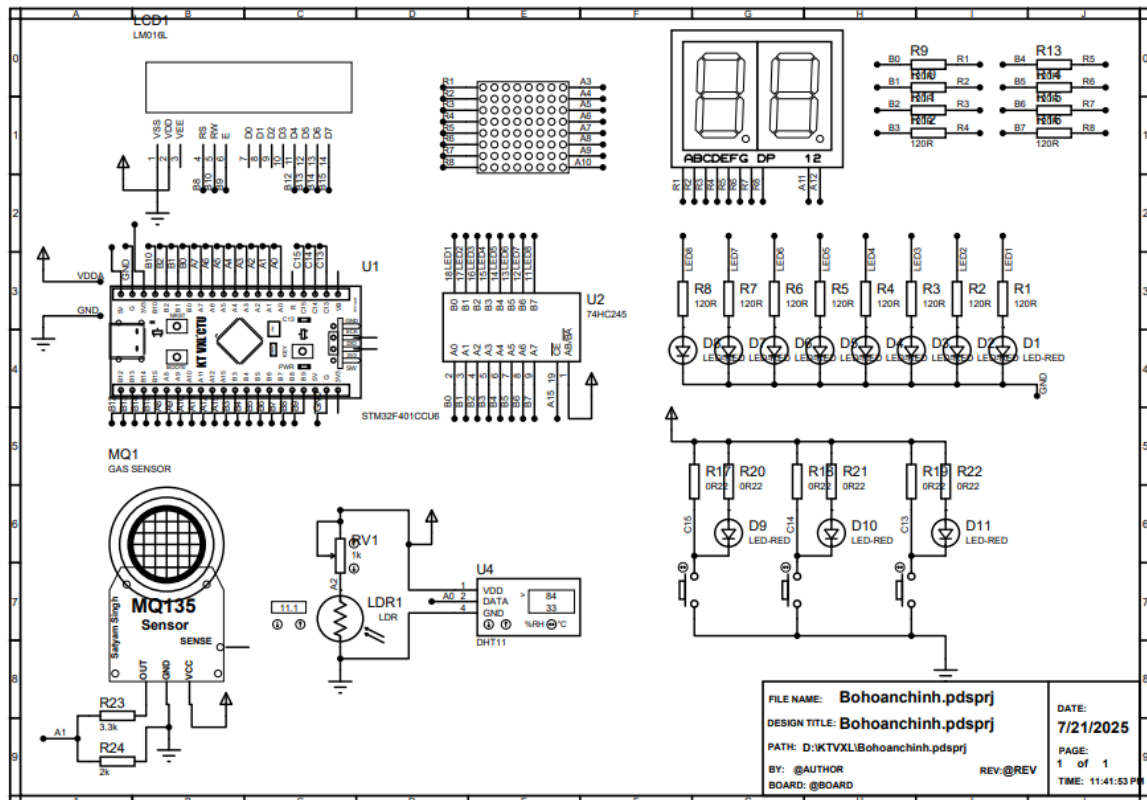


Ngõ ra và ngõ vào của hệ thống:

**Ngõ vào:** tín hiệu từ DHT11 (1-wire), MQ135 (ADC), LDR (ADC), 3 nút nhấn (GPIO/EXTI).

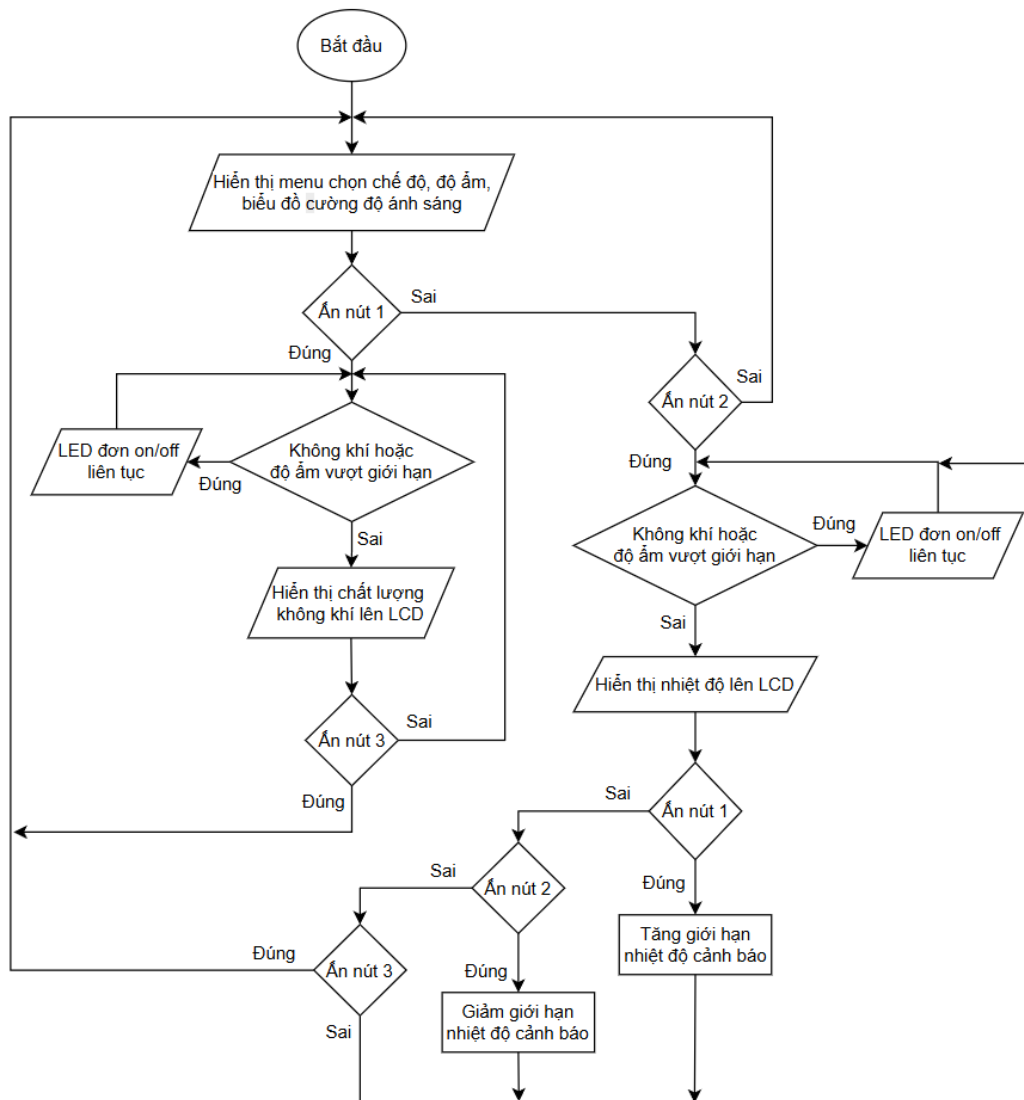
**Ngõ ra:** LCD (GPIO), LED ma trận (GPIO quét), LED 7 đoạn đôi (GPIO quét), LED đơn (GPIO output).

Sơ đồ mạch nguyên lý của hệ thống:

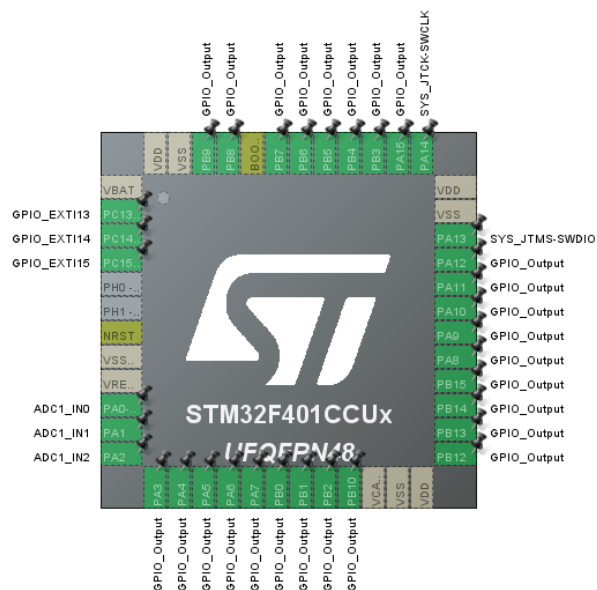




## Lưu đồ giải thuật của điều khiển cho hệ thống:



## Cấu hình ngõ vào ngõ ra trên phần mềm nhúng:





---

## Viết chương trình nhúng:

```
#include "main.h"
#include "lcd_16x2.h"
int mode = 0;
int giatri_LDR = 0;
int giatri_MQ135 = 0;
int ledMatrix[8];
int ldr_matrix[8][8] = {
    {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80},
    {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0xC0},
    {0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x60, 0xE0},
    {0x00, 0x00, 0x00, 0x00, 0x10, 0x30, 0x70, 0xF0},
    {0x00, 0x00, 0x00, 0x08, 0x18, 0x38, 0x78, 0xF8},
    {0x00, 0x00, 0x04, 0x0C, 0x1C, 0x3C, 0x7C, 0xFC},
    {0x00, 0x02, 0x06, 0x0E, 0x1E, 0x3E, 0x7E, 0xFE},
    {0x01, 0x03, 0x07, 0x0F, 0x1F, 0x3F, 0x7F, 0xFF}
};
int maled[10] = {
    0b00111111, // 0
    0b00000110, // 1
    0b01011011, // 2
    0b01001111, // 3
    0b01100110, // 4
    0b01101101, // 5
    0b01111101, // 6
    0b00000111, // 7
    0b01111111, // 8
    0b01101111 // 9
};
// Khởi tạo delay mili giây từ TIMER
void Delay_us(uint16_t us) {
    __HAL_TIM_SET_COUNTER(&htim2, 0); // Reset bộ đếm
    HAL_TIM_Base_Start(&htim2);      // Bắt đầu Timer
    while (__HAL_TIM_GET_COUNTER(&htim2) < us);
    HAL_TIM_Base_Stop(&htim2);        // Dừng Timer nếu muốn
}
// Định nghĩa lại chân A0 thành ngoVao
void ngoVao() {
    GPIO_InitTypeDef GPIO_InitStruct = {0};
    GPIO_InitStruct.Pin = GPIO_PIN_0;
    GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
```

```
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
}
// Định nghĩa lại chân A1 thành ngoRa
void ngoRa() {
    GPIO_InitTypeDef GPIO_InitStruct = {0};
    GPIO_InitStruct.Pin = GPIO_PIN_0;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_PULLUP;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
}
// Bắt đầu khởi chạy DHT11
void batDauDHT11() {
    ngoRa();
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, SET);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, RESET);
    HAL_Delay(18);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, SET);
    Delay_us(20);
    //Đoi phan hoi tu DHT11
    ngoVao();
    while (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) ==
1) {} //Đoi xuống 0
    while (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) ==
0) {} //Đoi lên 1
    while (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) ==
1) {} //Đoi xuống 0
    //Khi đã xuống 0 thì 4 bit bắt đầu chuyển
}
//Đọc mỗi lần 8 bit (1byte)
int docGiaTriDHT11() {
    int value = 0;
    for (int i = 0; i < 8; i++) {
        while (HAL_GPIO_ReadPin(GPIOA,
GPIO_PIN_0) == 0) {} //Đoi lên 1
        Delay_us(50);
        if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)
== 1) {
            //Nếu đã hết 50us nó vẫn còn ở mức 2 thì là
bit1
            value = (value << 1) | (1 << 0);
        } else {
            value = (value << 1) & ~(1 << 0);
```



```

    }
    while (HAL_GPIO_ReadPin(GPIOA,
GPIO_PIN_0) == 1) {};//Đoi xuống 0
    }
    return value;
}

// Đọc giá trị kênh ADC
uint16_t docADC(uint32_t channel) {
    ADC_ChannelConfTypeDef sConfig = {0};
    sConfig.Channel = channel;
    sConfig.Rank = 1;
    sConfig.SamplingTime =
ADC_SAMPLETIME_3CYCLES;

    HAL_ADC_ConfigChannel(&hadc1, &sConfig);
    HAL_ADC_Start(&hadc1);
    HAL_ADC_PollForConversion(&hadc1,
HAL_MAX_DELAY);
    uint16_t value = HAL_ADC_GetValue(&hadc1);
    HAL_ADC_Stop(&hadc1);

    return value;
}

int nhietdo_nguyen = 0, doam_nguyen = 0;
float nhietdo_thapphan = 0, doam_thapphan = 0;
uint8_t kiểmtra = 0;
int nhietdo_canhbao = 40;
//Đọc dữ liệu
void docDuLieuDHT11() {
    batDauDHT11();
    doam_nguyen = docGiaTriDHT11();
    nhietdo_thapphan = docGiaTriDHT11();
    nhietdo_nguyen = docGiaTriDHT11();
    doam_thapphan = docGiaTriDHT11();
    kiểmtra = docGiaTriDHT11();
}

//Chân row (PB0 -> PB7)
#define ROW1_Pin GPIO_PIN_0
#define ROW1_GPIO_Port GPIOB
#define ROW2_Pin GPIO_PIN_1
#define ROW2_GPIO_Port GPIOB
#define ROW3_Pin GPIO_PIN_2
#define ROW3_GPIO_Port GPIOB
#define ROW4_Pin GPIO_PIN_3
#define ROW4_GPIO_Port GPIOB

```

```

#define ROW5_Pin GPIO_PIN_4
#define ROW5_GPIO_Port GPIOB
#define ROW6_Pin GPIO_PIN_5
#define ROW6_GPIO_Port GPIOB
#define ROW7_Pin GPIO_PIN_6
#define ROW7_GPIO_Port GPIOB
#define ROW8_Pin GPIO_PIN_7
#define ROW8_GPIO_Port GPIOB
//Chân column (PA3 -> PA10)
#define COL1_Pin GPIO_PIN_3
#define COL1_GPIO_Port GPIOA
#define COL2_Pin GPIO_PIN_4
#define COL2_GPIO_Port GPIOA
#define COL3_Pin GPIO_PIN_5
#define COL3_GPIO_Port GPIOA
#define COL4_Pin GPIO_PIN_6
#define COL4_GPIO_Port GPIOA
#define COL5_Pin GPIO_PIN_7
#define COL5_GPIO_Port GPIOA
#define COL6_Pin GPIO_PIN_8
#define COL6_GPIO_Port GPIOA
#define COL7_Pin GPIO_PIN_9
#define COL7_GPIO_Port GPIOA
#define COL8_Pin GPIO_PIN_10
#define COL8_GPIO_Port GPIOA
//Mảng lưu trữ các chân hàng
GPIO_TypeDef* rowPorts[8] = {
    ROW1_GPIO_Port, ROW2_GPIO_Port,
    ROW3_GPIO_Port, ROW4_GPIO_Port,
    ROW5_GPIO_Port, ROW6_GPIO_Port,
    ROW7_GPIO_Port, ROW8_GPIO_Port
};
uint16_t rowPins[8] = {
    ROW1_Pin, ROW2_Pin, ROW3_Pin, ROW4_Pin,
    ROW5_Pin, ROW6_Pin, ROW7_Pin, ROW8_Pin
};
// Mảng lưu trữ các chân cột
GPIO_TypeDef* colPorts[8] = {
    COL1_GPIO_Port, COL2_GPIO_Port,
    COL3_GPIO_Port, COL4_GPIO_Port,
    COL5_GPIO_Port, COL6_GPIO_Port,
    COL7_GPIO_Port, COL8_GPIO_Port
};
uint16_t colPins[8] = {
    COL1_Pin, COL2_Pin, COL3_Pin, COL4_Pin,

```



```

COL5_Pin, COL6_Pin, COL7_Pin, COL8_Pin
};
// Hàm quét LED matrix
void scanLedMatrix() {
    for (int i = 0; i < 24; i++) {
        for (int row = 0; row < 8; row++) {
            // Bật hàng
            HAL_GPIO_WritePin(rowPorts[row],
rowPins[row], 1);
            // Đặt dữ liệu cột
            for (int col = 0; col < 8; col++) {
                if ((ledMatrix[row] >> (7 - col)) & 0x01) { //7-
col để đảo theo trục Y
                    HAL_GPIO_WritePin(colPorts[col],
colPins[col], 0); // Âm chung: Bật LED
                } else {
                    HAL_GPIO_WritePin(colPorts[col],
colPins[col], 1); // Âm chung: Tắt LED
                }
            }
            // Tắt hàng và cột
            HAL_GPIO_WritePin(rowPorts[row],
rowPins[row], 0);
            for (int col = 0; col < 8; col++) {
                HAL_GPIO_WritePin(colPorts[col],
colPins[col], 1);
            }
        }
    }
}
//Led 7 đoạn
void hienthi(int x) {
    uint8_t chuc = 0;
    uint8_t donvi = 0;
    chuc = x / 10;
    donvi = x % 10;
    //Hang don vi
    HAL_GPIO_WritePin(GPIOB, maled[donvi], 1);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, 1);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, 0);
    HAL_Delay(0);
    HAL_GPIO_WritePin(GPIOB, maled[donvi], 0);
    //Hang chuc
    HAL_GPIO_WritePin(GPIOB, maled[chuc], 1);
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, 0);

```

```

    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, 1);
    HAL_Delay(0);
    HAL_GPIO_WritePin(GPIOB, maled[chuc], 0);
}
// Hien thi cung luc LED 7 va LED ma tran
void scan_Matrix_7(int i) {
    for (int j = 0; j < 6; j++) {
        // Hiện thị LED 7 đoạn trước
        hienthi(i); // Hiện thị số i
        // Tạm tắt LED 7 đoạn để không chổng tín hiệu
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, 1);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, 1);

        // Quét LED ma trận 1 lần
        scanLedMatrix();
    }
}
// Menu
void mode_0() {
    Lcd_clear_display();
    Lcd_gotoxy(0, 0);
    Lcd_write_string("1. Khong khi");
    Lcd_gotoxy(0, 1);
    Lcd_write_string("2. Nhiet do");
}
// Xu ly nut nhan
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) {
    switch (GPIO_Pin) {
        case GPIO_PIN_13: // Nút 1
            if (mode == 0)
                mode = 1;
            else if (mode == 2)
                nhietdo_canhbao++; // Tăng ngưỡng
            break;
        case GPIO_PIN_14: // Nút 2
            if (mode == 0)
                mode = 2;
            else if (mode == 2 && nhietdo_canhbao > 0)
                nhietdo_canhbao--; // Giảm ngưỡng
            break;
        case GPIO_PIN_15: // Nút 3
            mode = 0;
            break;
    }
}

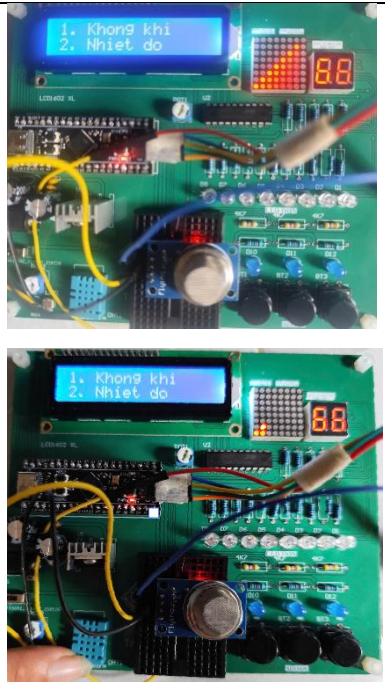
```



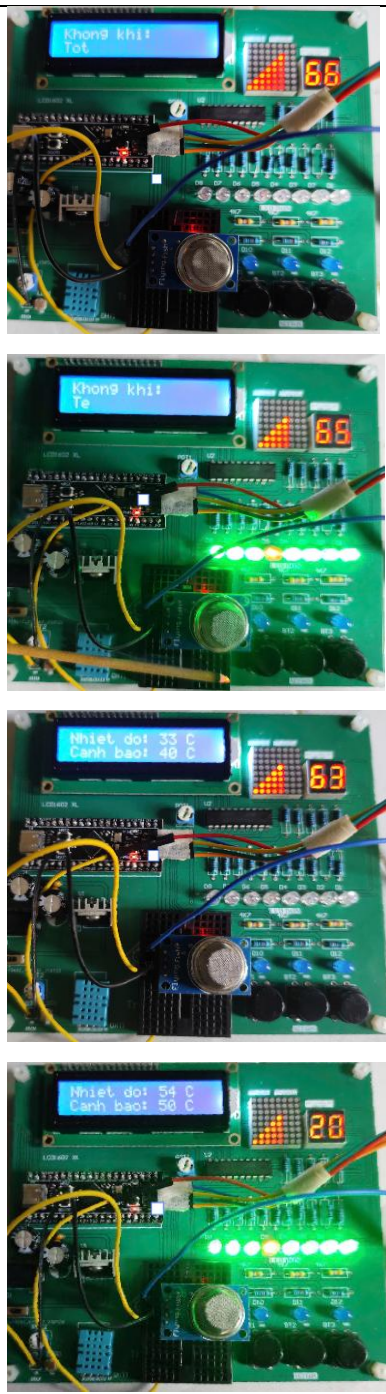
<pre> // Cap nhat LED ma tran theo LDR void update_Matrix2LDR() {     int index = (giatri_LDR - 500) * 7 / (3500 - 500);     index = 7 - index;     if (index &lt; 0) index = 0;     if (index &gt; 7) index = 7;     for (int i = 0; i &lt; 8; i++) {         ledMatrix[i] = ldr_matrix[index][i];     } }  int main(void) {     Lcd_Init();     Lcd_clear_display();     docDuLieuDHT11(); // Đọc DHT11 lần đầu     while (1) {         giatri_MQ135 = docADC(ADC_CHANNEL_1);         giatri_LDR = docADC(ADC_CHANNEL_2);         scan_Matrix_7(doam_nguyen);         update_Matrix2LDR();         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 1);         if (nhietdo_nguyen &gt;= nhietdo_canhbao    giatri_MQ135 &gt;= 3000) {             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 0);         }         if (mode == 0) {             Lcd_gotoxy(0, 0);             Lcd_write_string("1. Khong khi ");             Lcd_gotoxy(0, 1);             Lcd_write_string("2. Nhiet do ");         } else if (mode == 1) {             Lcd_gotoxy(0, 0);             Lcd_write_string("Khong khi: ");             if (giatri_MQ135 &lt; 3000) {                 Lcd_gotoxy(0, 1);                 Lcd_write_string("Tot ");                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 1); </pre>	<pre>         if (nhietdo_nguyen &gt;= nhietdo_canhbao) {             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 0);         }         } else {             Lcd_gotoxy(0, 1);             Lcd_write_string("Te ");             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 0);         }         } else if (mode == 2) {             docDuLieuDHT11();             Lcd_gotoxy(0, 0);             Lcd_write_string("Nhiet do: ");             Lcd_gotoxy(10, 0);             Lcd_write_int(nhietdo_nguyen);             Lcd_gotoxy(13, 0);             Lcd_write_string("C ");             Lcd_gotoxy(0, 1);             Lcd_write_string("Canh bao: ");             Lcd_gotoxy(10, 1);             Lcd_write_int(nhietdo_canhbao);             Lcd_gotoxy(13, 1);             Lcd_write_string("C ");             if (nhietdo_nguyen &gt;= nhietdo_canhbao    giatri_MQ135 &gt;= 3000) {                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 0);             }             } else {                 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 1);             }         }     } } </pre>
---	---



## Kiểm tra giải thuật và hoạt động của phần cứng:

	Giải thuật	Code	Kết quả
-LCD -LED ma trận -LED 7 đoạn	-LCD hiển thị menu (mode 0). -LED ma trận hiển thị cột sáng từ LDR. -LED 7 đoạn hiển thị giá trị độ ẩm.	<pre> // Hàm quét LED matrix void scanLedMatrix() {     for (int i = 0; i &lt; 24; i++) {         for (int row = 0; row &lt; 8; row++) {             // Bật hàng             HAL_GPIO_WritePin(rowPorts[row], rowPins[row], 1);             // Duyệt dữ liệu cột             for (int col = 0; col &lt; 8; col++) {                 if ((ledMatrix[row] &gt;&gt; (7 - col)) &amp; 0x01) {                     //7-col để đảo theo trục Y                     HAL_GPIO_WritePin(colPorts[col], colPins[col], 0);                     // Âm chung: Bật LED                 } else {                     HAL_GPIO_WritePin(colPorts[col], colPins[col], 1);                     // Âm chung: Tắt LED                 }             }             // Tắt hàng và cột             HAL_GPIO_WritePin(rowPorts[row], rowPins[row], 0);             for (int col = 0; col &lt; 8; col++) {                 HAL_GPIO_WritePin(colPorts[col], colPins[col], 1);             }         }     } }  //Led 7 đoạn void hienthi(int x) {     uint8_t chuc = 0;     uint8_t donvi = 0;     chuc = x / 10;     donvi = x % 10;     //Hang don vi     HAL_GPIO_WritePin(GPIOB, maled[donvi], 1);     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, 1);     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, 0);     HAL_Delay(0);     HAL_GPIO_WritePin(GPIOB, maled[donvi], 0);     //Hang chuc     HAL_GPIO_WritePin(GPIOB, maled[chuc], 1);     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, 0);     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, 1);     HAL_Delay(0);     HAL_GPIO_WritePin(GPIOB, maled[chuc], 0); }  // Hien thi cung luc LED 7 va LED ma tran void scan_Matrix_7(int i) {     for (int j = 0; j &lt; 6; j++) {         // Hiển thị LED 7 đoạn trước         hienthi(i); // Hiển thị số i         // Tạm tắt LED 7 đoạn để không chòng tén hiệu         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, 1);         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, 1);         // Quét LED ma trận 1 lần         scanLedMatrix();     } }  // Menu void mode_0() {     Lcd_clear_display();     Lcd_gotoxy(0, 0);     Lcd_write_string("1. Khong khi");     Lcd_gotoxy(0, 1);     Lcd_write_string("2. Nhiet do"); } </pre>	



		<pre>// Cap nhat LED ma tran theo LDR void update_Matrix2LDR() {     int index = (giatri_LDR - 500) * 7 / (3500 - 500);     index = 7 - index;      if (index &lt; 0) index = 0;     if (index &gt; 7) index = 7;      for (int i = 0; i &lt; 8; i++) {         ledMatrix[i] = ldr_matrix[index][i];     } }</pre>	
DHT11 MQ135	<p>- Khi ở mode 1: Hiển thị nhiệt độ lên LCD và cảnh báo LED đơn khi vượt ngưỡng.</p> <p>- Khi ở mode 2: Hiển thị chất lượng không khí lên LCD và cảnh báo Led đơn khi chất lượng tệ.</p>	<pre>giatri_MQ135 = docADC(ADC_CHANNEL_1); giatri_LDR = docADC(ADC_CHANNEL_2); HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 1); if (nhietdo_nguyen &gt;= nhietdo_canhbao    giatri_MQ135 &gt;= 3000) {     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 0); } if (mode == 0) {     Lcd_gotoxy(0, 0);     Lcd_write_string("1. Khong khi    ");     Lcd_gotoxy(0, 1);     Lcd_write_string("2. Nhiet do    "); } else if (mode == 1) {     Lcd_gotoxy(0, 0);     Lcd_write_string("Khong khi:    ");     if (giatri_MQ135 &lt; 3000) {         Lcd_gotoxy(0, 1);         Lcd_write_string("Tot    ");         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 1);         if (nhietdo_nguyen &gt;= nhietdo_canhbao)             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 0);     } else {         Lcd_gotoxy(0, 1);         Lcd_write_string("Te    ");         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 0);     } } else if (mode == 2) {     read_full_data();     Lcd_gotoxy(0, 0);     Lcd_write_string("Nhiet do:    ");     Lcd_gotoxy(10, 0);     Lcd_write_int(nhietdo_nguyen);     Lcd_gotoxy(13, 0);     Lcd_write_string("C    ");     Lcd_gotoxy(0, 1);     Lcd_write_string("Canh bao:    ");     Lcd_gotoxy(10, 1);     Lcd_write_int(nhietdo_canhbao);     Lcd_gotoxy(13, 1);     Lcd_write_string("C    ");     if (nhietdo_nguyen &gt;= nhietdo_canhbao    giatri_MQ135 &gt;= 3000) {         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 0);     } else {         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 1);     } }</pre>	

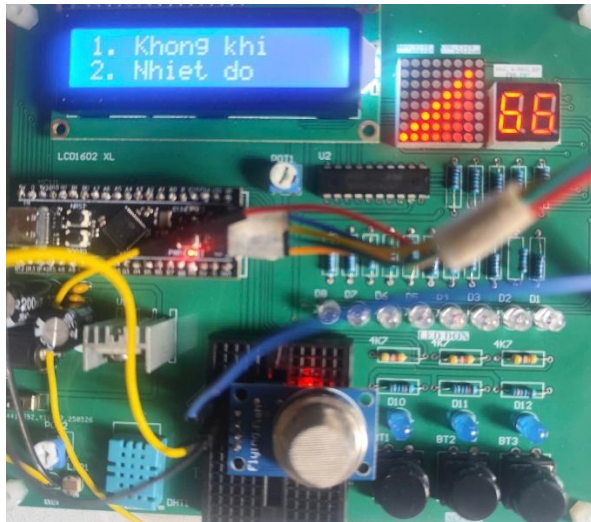


---

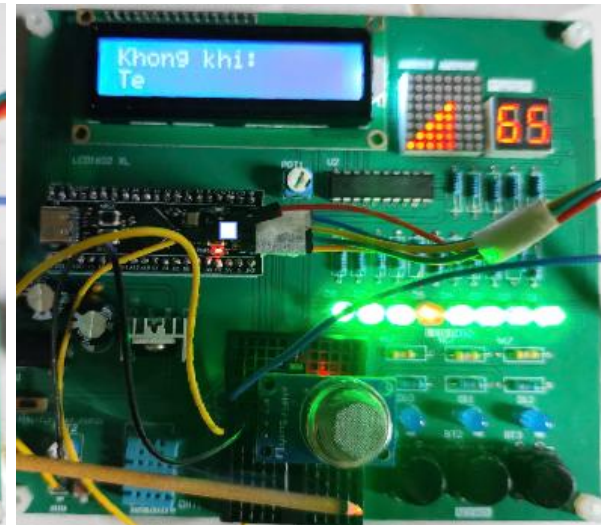
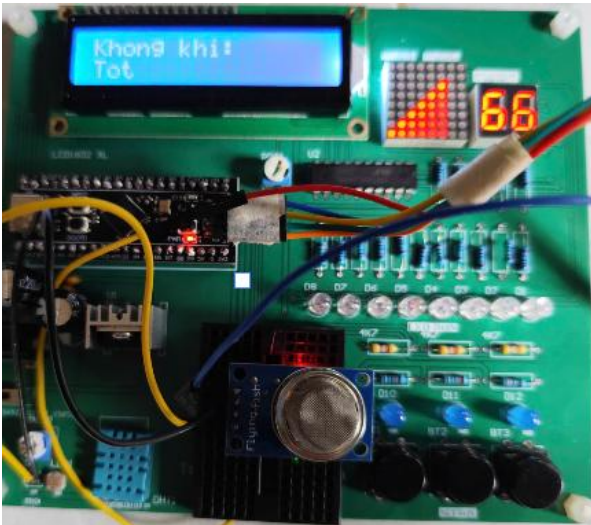
## 2.2. Kết quả đạt được

### Kết quả thực tế trên board:

Sử dụng LED từ điện thoại hoặc tay che để tăng/giảm cột sáng của LED ma trận.



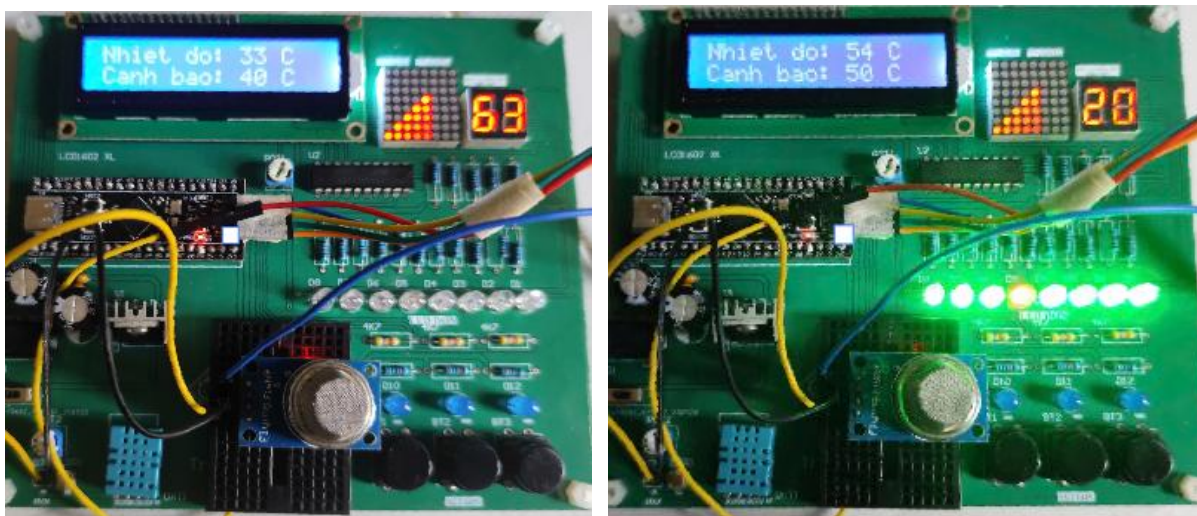
Sử dụng khói từ nhang, giấy, khí gas (bật lửa) để giảm chất lượng không khí (tệ).





---

Sử dụng máy sấy để tăng nhiệt độ và giảm độ ẩm.



### **Ưu điểm và nhược điểm của hệ thống:**

**Ưu điểm:** Chi phí thấp, linh kiện phổ biến, đáp ứng nhu cầu giám sát cơ bản trong nhà kính, giao diện điều khiển đơn giản, có chức năng cảnh báo.

**Nhược điểm:** Giao diện còn hạn chế, độ chính xác và độ bền của cảm biến còn hạn chế, không có tính năng kết nối để điều khiển từ xa, thiếu tính năng điều khiển tự động.

### **Khả năng phát triển trong tương lai của hệ thống:**

**Điều khiển tự động:** Tự động điều chỉnh lại môi trường khi có sự thay đổi: Phát triển giải thuật để điều khiển quạt, máy sưởi, đèn chiếu sáng,... dựa trên giá trị cảm biến, điều khiển theo ngưỡng thông minh: tự động cập nhật ngưỡng tối ưu theo thời gian thay vì đặt ngưỡng cố định.

**Giao tiếp và điều khiển từ xa:** Thêm module Wi-Fi hoặc Bluetooth: phát triển thuật toán giao tiếp để gửi dữ liệu lên điện thoại và điều khiển từ xa qua ứng dụng di động/web: Viết phần mềm nhận/gửi lệnh từ điện thoại.

**Quản lý thời gian và dữ liệu:** Ghi dữ liệu vào thẻ nhớ kết hợp với thuật toán phân tích dữ liệu cũ để đưa ra lời khuyên cần thiết, thêm lịch trình định kỳ (VD: dựa vào nhiệt độ hằng ngày tự cài đặt thời gian tưới cây).



---

## CHƯƠNG III: KẾT LUẬN

Mạch hiển thị được các thông số môi trường rõ ràng như nhiệt độ, độ ẩm, chất lượng không khí và cường độ ánh sáng. Khi vượt ngưỡng sẽ có cảnh báo bằng đèn LED. Mặc dù mạch vẫn còn nhiều hạn chế như: Giao diện còn đơn giản, hệ thống còn hạn chế khi LED hiển thị bị chớp tắt liên tục, chưa cảnh báo được khi cường độ ánh sáng và độ ẩm vượt ngưỡng, nhưng đã làm tốt mục tiêu đề ra ban đầu.

Mô hình giám sát môi trường có ý nghĩa đặc biệt trong nông nghiệp, trong các khu cần kiểm soát thông số môi trường khắc khe. Góp phần vào việc hiện đại hóa hoạt động sản xuất nông nghiệp.

Trong tương lai, hệ thống có thể được phát triển theo hướng tự động hóa và kết nối mạng: bổ sung chức năng điều khiển thiết bị ngoại vi, lưu trữ, phân tích dữ liệu, giao tiếp không dây,...

## TÀI LIỆU THAM KHẢO

- [1] Tài liệu Kỹ thuật vi xử lý do giảng viên cung cấp
- [2] Tài liệu về code STM32 từ ChatGPT, GitHub
- [3] Tài liệu về linh kiện [alldatasheet.com](http://alldatasheet.com)