

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**KHOA ĐIỆN TỬ VIỄN THÔNG**



**BÁO CÁO ĐỒ ÁN**

**Môn học: THỰC HÀNH VI ĐIỀU KHIỂN**

**Giảng viên: Trần Tuấn Kiệt**

**ĐỀ TÀI: THIẾT BỊ ĐẾM NGƯỜI RA VÀO PHÒNG.**

*TP. Hồ Chí Minh, ngày 16 tháng 7 năm 2022*

## Contents

<b>MỞ ĐẦU</b> .....	2
<b>1.1. Giới Thiệu Về Các Linh Kiện Sử Dụng</b> .....	4
<b>1.1.1. Pic18f4550</b> .....	4
1.1.1.1. Thông số kỹ thuật PIC18F4550:.....	4
Hình ảnh minh họa: .....	4
.....	4
1.1.1.2. Sơ đồ chân của PIC18F4550 .....	6
.....	6
<b>1.1.2. Màn hình LCD:</b> .....	7
Hình 7: Sơ đồ chân LCD.....	7
<b>1.1.3. Cảm biến vật cản hồng ngoại:</b> .....	8
<b>1.1.4. Các linh kiện khác:</b> .....	10
1.1.4.1. Tụ điện không phân cực:.....	10
1.1.4.2. Điện trở:.....	11
1.1.4.3. Thạch anh: .....	11
1.1.4.4. Biến trở: .....	11
1.1.4.5. LED:.....	11
1.1.4.6. Hàng rào: .....	11
1.1.4.7. TBLOCK12: .....	11
<b>CHƯƠNG 2: NGUYÊN LÝ HOẠT ĐỘNG</b> .....	15
2.1 Lưu đồ hoạt động và nguyên lý:.....	15
2.2 Sơ đồ nguyên lý:.....	15
2.3 Nguyên lý hoạt động của cảm biến hồng ngoại: .....	15
<b>CHƯƠNG 3: Chương Trình Phần Mềm Sử Dụng Hệ Thông</b> .....	17
3.1 Sơ lược về phần mềm sử dụng:.....	17
<b>KẾT LUẬN</b> .....	30

## ***MỞ ĐẦU***

Thế kỷ 21 mở ra một thời đại mới, thời đại khoa học công nghệ đòi hỏi con người luôn luôn không ngừng tìm tòi học hỏi để phát triển và tiến bộ. Với sự nhảy vọt của khoa học, kỹ thuật điện- điện tử, mà vì thế trong một thời gian ngắn nó đã đạt được những thành tựu to lớn trong hầu hết các lĩnh vực trong đời sống xã hội. Thiết bị và công nghệ ngày càng đổi mới để góp phần nâng cao chất lượng cuộc sống. Ngày nay các thiết bị vi điều khiển có ứng dụng càng rộng rãi với ưu điểm nhỏ gọn, linh hoạt và có thể điều khiển được rộng rãi. Vi điều khiển ngày càng chiếm lĩnh và đóng vai trò cực kỳ quan trọng trong kỹ thuật điều khiển và tự động hóa.

Giờ đây, nhu cầu chuyên dụng hóa, tối ưu (thời gian, không gian, giá thành) bảo mật, tính chủ động linh hoạt trong công nghệ... ngày càng đòi hỏi khắc khe việc đưa ra công nghệ mới trong lĩnh vực chế tạo mạch điều khiển điện tử, để đáp ứng các nhu cầu cấp thiết trong khoa học kỹ thuật điện -điện tử. Kỹ thuật vi điều khiển hiện nay rất phát triển, nó đáp ứng được nhu cầu của nhiều ngành lĩnh vực sản xuất công nghiệp, tự động hóa, trong đời sống...So với kỹ thuật số thì kỹ thuật vi điều khiển nhỏ gọn hơn, do đó nó được tập hợp lại và có khả năng lập trình để điều khiển nên tiện dụng và cơ động. Với các tính chất ưu việt đó, trong đề tài này em sử dụng vi điều khiển thiết bị đếm người ra vào phòng và đồng thời hiển thị lên LCD.Đề tài này thiết kế dựa trên kiến thức đã học, sách tham khảo và một số nguồn tài liệu khác.

Nhóm chúng em xin gửi lời cảm ơn sâu sắc đến thầy **Trần Tuấn Kiệt** đã nhiệt tình hướng dẫn, giúp đỡ và tạo những điều kiện tốt nhất để nhóm em hoàn thành đồ án này. Tuy nhiên do thời gian và trình độ có hạn nên em không tránh khỏi những sai sót. Vì vậy mong thầy, góp ý xây dựng giúp đỡ để hoàn thành đề tài.

***TP.HCM, ngày 16 tháng 7 năm 2022***

## ***TỔNG QUÁT***

***Chương 1: Tổng quan về mạch điều khiển sử dụng cảm biến vật cản hồng ngoại và pic18f4550.***

***Chương 2: Nguyên lý hoạt động.***

***Chương 3: Chương trình phần mềm sử dụng cho hệ thống.***

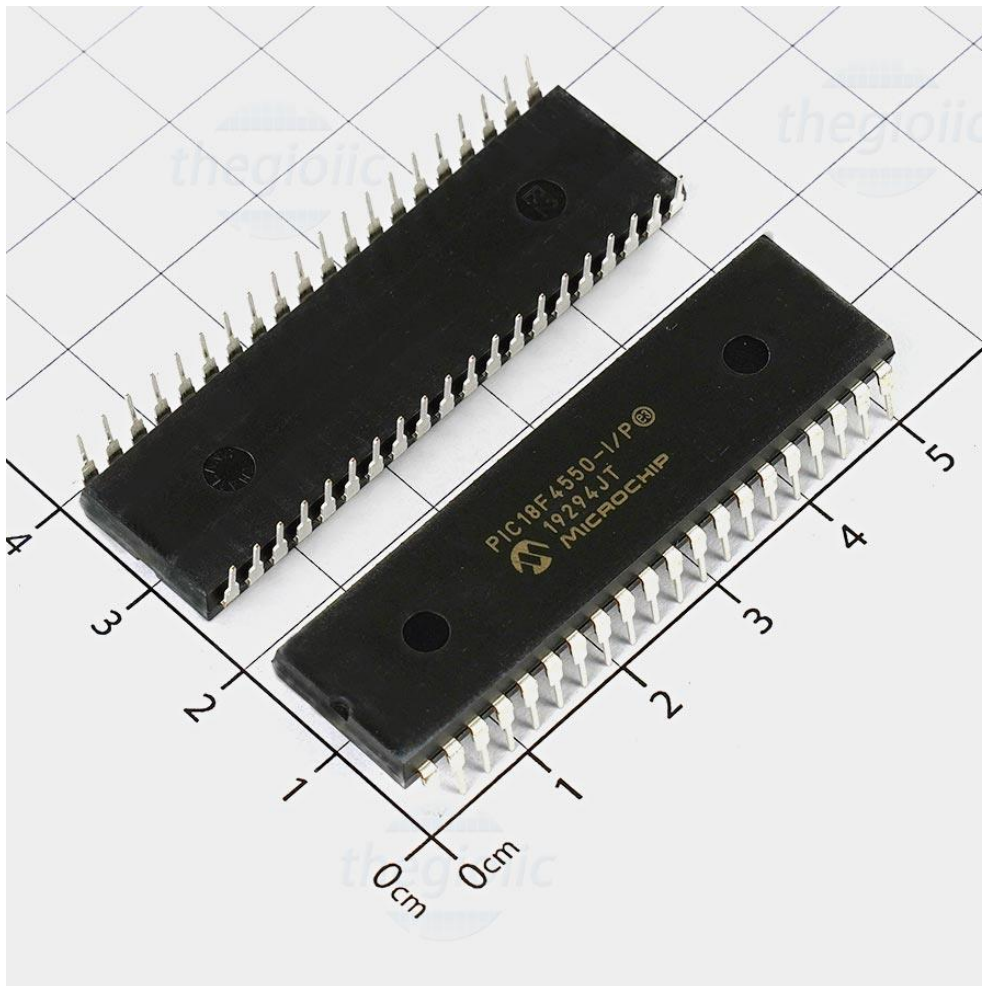
# CHƯƠNG 1: TỔNG QUAN VỀ MẠCH ĐIỀU KHIỂN SỬ DỤNG CẢM BIẾN VẬT CẢN HỒNG NGOẠI VÀ PIC18F4550

## 1.1. Giới Thiệu Về Các Linh Kiện Sử Dụng

### 1.1.1. Pic18f4550

#### 1.1.1.1. Thông số kỹ thuật PIC18F4550:

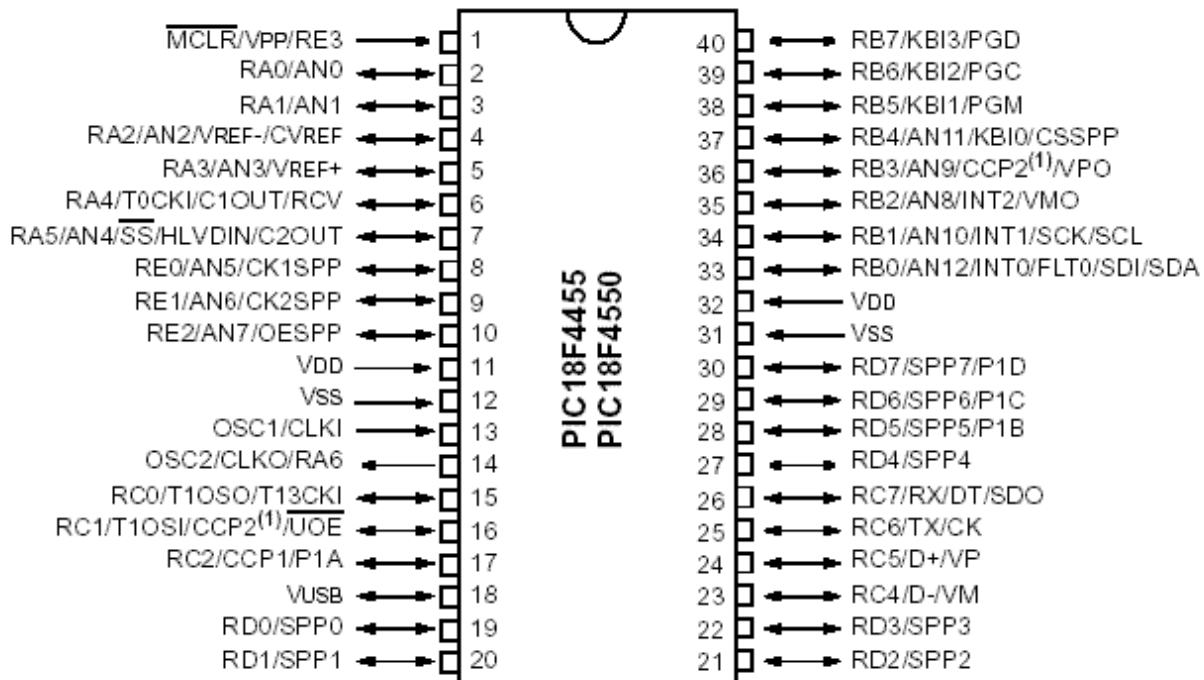
Hình ảnh minh họa:



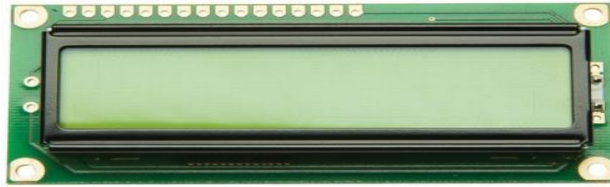
Standard Package	27
Category	Integrated Circuits (ICs)
Family	Embedded – Microcontrollers
Series	PIC® 18F
Packaging	Tube
Core Processor	PIC
Core Size	8-Bit
Speed	48MHz
Connectivity	PC, SPI, UART / USART, USB
Peripherals	Brown-out Detector / Reset, HLVD, POR, PWM, WDT
Number of I /O	24
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8

RAM Size	2K x 8
Voltage – Supply (Vcc/Vdd)	4.2 V ~ 5.5 V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C
Package / Case	28-SOIC (0.295", 7.50mm Width)
Other Names	PIC18F4550

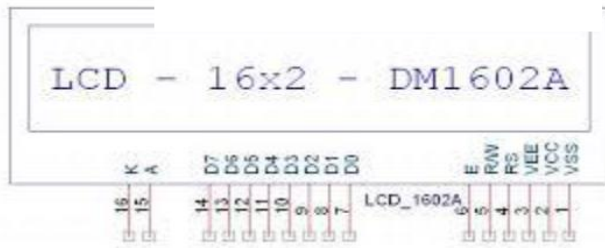
#### 1.1.1.2. Sơ đồ chân của PIC18F4550



### 1.1.2. Màn hình LCD:



Hình 6: LCD 16X2



Hình 7: Sơ đồ chân LCD

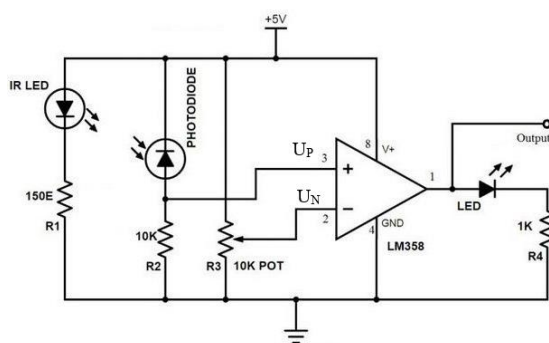
#### ○ Chức năng các chân:

Chân	Ký hiệu	Mô tả
1	$V_{SS}$	Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển
2	$V_{DD}$	Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với $V_{CC}=5V$ của mạch điều khiển
3	$V_{EE}$	Điều chỉnh độ tương phản của LCD.
4	$RS$	Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” ( $V_{CC}$ ) để chọn thanh ghi. + Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read) + Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD.



5	R/W	Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc.
6	E	Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E. + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào(chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E. + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát
7 - 14	DB0 - DB7	Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này : + Chế độ 8 bit : Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7. + Chế độ 4 bit : Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7
15	-	Nguồn dương cho đèn nền
16	-	GND cho đèn nền

### 1.1.3. Cảm biến vật cản hồng ngoại:



Hình 2.0. 1 Mạch nguyên lý



Hình 2.0. 2 Hình dạng thực tế

**Trong chế độ trực tiếp:** đèn LED hồng ngoại và điốt quang được giữ ở phía trước của nhau, do đó bức xạ hồng ngoại có thể trực tiếp rơi vào điốt quang. Nếu chúng ta đặt bất kỳ vật thể nào giữa chúng, thì nó sẽ ngăn chặn sự chiếu xuống của ánh sáng IR trên điốt quang.

**Trong chế độ gián tiếp:** cả đèn LED hồng ngoại và diode quang đều được đặt song song (cạnh nhau), hướng cả hai về cùng một hướng. Khi một vật được giữ trước cặp LED thu phát hồng ngoại IR, ánh sáng hồng ngoại sẽ bị vật thể phản xạ và bị photodiode hấp thụ. Lưu ý rằng vật thể không được có màu đen vì nó sẽ hấp thụ tất cả ánh sáng hồng ngoại IR, thay vì phản xạ. Nói chung, cặp LED thu phát hồng ngoại IR được đặt theo kiểu này trong Mô-đun cảm biến IR.

Để xây dựng một Module cảm biến hồng ngoại IR, chủ yếu cần một cặp LED thu phát hồng ngoại và IC LM358 với một số điện trở và đèn LED.

### **LED phát hồng ngoại (IR LED)**

IR LED phát ra ánh sáng, trong dải tần số hồng ngoại. Chúng ta không nhìn thấy ánh sáng hồng ngoại vì bước sóng của nó (700nm – 1mm) cao hơn nhiều so với dải ánh sáng nhìn thấy. Mọi thứ sinh ra nhiệt đều phát ra tia hồng ngoại như cơ thể con người chúng ta. Tia hồng ngoại có các đặc tính giống như ánh sáng nhìn thấy, như nó có thể được hội tụ, phản xạ và phân cực như ánh sáng nhìn thấy.



*Hình 2.0. 3 LED phát hồng ngoại*

IR LED hoạt động như một đèn LED bình thường, nó tiêu thụ dòng điện 20mA và nguồn điện 3V. Đèn LED hồng ngoại có góc phát ánh sáng xấp xỉ 20-60 độ và phạm vi khoảng vài cm đến vài feet, nó phụ thuộc vào loại máy phát IR và nhà sản xuất.

### **LED thu hồng ngoại ( photodiode)**

Điốt quang được coi là điện trở phụ thuộc vào ánh sáng (LDR), có nghĩa là nó có điện trở rất cao khi thiếu ánh sáng và điện trở nhỏ yếu khi ánh sáng chiếu vào, điốt quang là một chất bán dẫn có tiếp giáp PN, hoạt động theo phân cực ngược (dẫn dòng điện theo hướng ngược lại khi có ánh sáng chiếu vào).



*Hình 2.0. 4 led thu phát hồng ngoại*

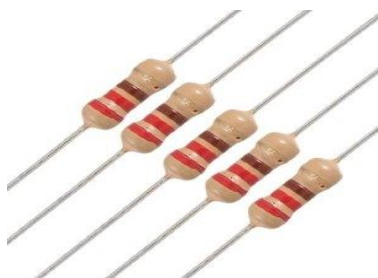
#### **1.1.4. Các linh kiện khác:**

##### **1.1.4.1. Tụ điện không phân cực:**



Tụ điện không phân cực: Đây là loại tụ không có quy định cực tính. Tụ này có điện dung nhỏ và dùng nhiều trong mạch điện có tần số cao hoặc mạch lọc nhiễu

#### 1.1.4.2. Điện trở:



Điện trở là một loại linh kiện điện tử thụ động có đặc tính cản trở dòng điện đi qua. Có tác dụng tạo ra sụt áp trên mạch khi mắc nối tiếp với

#### 1.1.4.3. Thạch anh:



Thạch anh sử dụng rất rộng rãi, hầu như ở đâu cũng có và giá thành thì nó cũng rất rẻ, khoảng Vài nghìn 1 con.

Ứng dụng của thạch anh trong điện tử đa phần để tạo ra tần số được ổn định vì tần số của thạch anh tạo ra rất ít bị ảnh hưởng bởi nhiệt độ hơn là các mạch dao động RC....

#### 1.1.4.4. Biến trở:



**Biến trở** là các thiết bị có [điện trở](#) thuần có thể biến đổi được theo ý muốn. Chúng có thể được sử dụng trong các mạch điện để điều chỉnh hoạt động của mạch điện.

Điện trở của thiết bị có thể được thay đổi bằng cách thay đổi [chiều dài](#) của dây dẫn điện trong thiết bị, hoặc bằng các tác động khác như [nhiệt độ](#) thay đổi, [ánh sáng](#) hoặc [bức xạ điện từ](#),...

Cấu tạo của biến trở gồm 2 thành phần chính là con chạy và cuộn dây được làm bằng hợp kim có điện trở suất lớn.

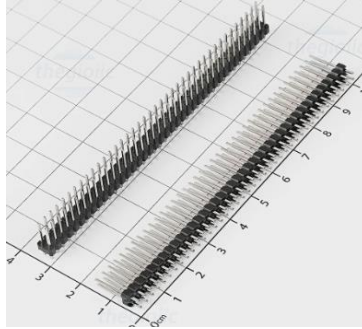
Biến trở thường ráp trong máy phục vụ cho quá trình sửa chữa, cân chỉnh của kỹ thuật viên.

#### 1.1.4.5. Led:



**LED** (viết tắt của *light-emitting diode*, có nghĩa là **diode phát sáng** hoặc **diode phát quang**) là các [diode](#) có khả năng phát ra [ánh sáng](#) hay tia [hồng ngoại](#), [tử ngoại](#). Cũng giống như diode, LED được cấu tạo từ một khối [bán dẫn loại p](#) ghép với một khối [bán dẫn loại n](#).

#### 1.1.4.6. Hàng rào (header)



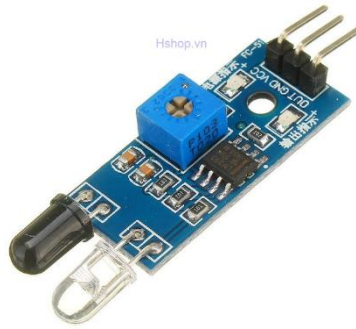
Rào đực và rào cái có tác dụng làm đế chân kết nối giữa mạch điện và dây nối, tránh dây nối hàn trực tiếp lên mạch có thể gây hư hỏng mạch điện của chúng ta.

#### 1.1.4.7. TBLOCK12:



TBLOCK12 là một khối được định dạng SBS được thiết kế để định vị chính xác các ống 15ML trên các hệ thống tự động. Nhãn vị trí chữ và số cung cấp theo dõi định hướng. Khối nhôm cung cấp khả năng dẫn nhiệt khi sử dụng với các thiết bị được kiểm soát nhiệt độ. Các tùy chỉnh có sẵn để phù hợp với nhiều loại ống và / hoặc để tối đa hóa diện tích bề mặt tiếp xúc để dẫn nhiệt tối ưu

#### 1.1.4.8. Cảm biến vật cản hồng ngoại IR Infrared Obstacle Avoidance :



Cảm biến vật cản hồng ngoại IR Infrared Obstacle Avoidance được sử dụng để nhận biết vật cản bằng ánh sáng hồng ngoại, cảm biến có cách sử dụng đơn giản với biến trở chỉnh khoảng cách nhận biết vật cản, ngõ ra dạng Digital dễ dàng giao tiếp và lập trình với Vi điều khiển, thích hợp để làm các ứng dụng Robot tránh vật cản, báo trộm, mô hình cửa tự động,...

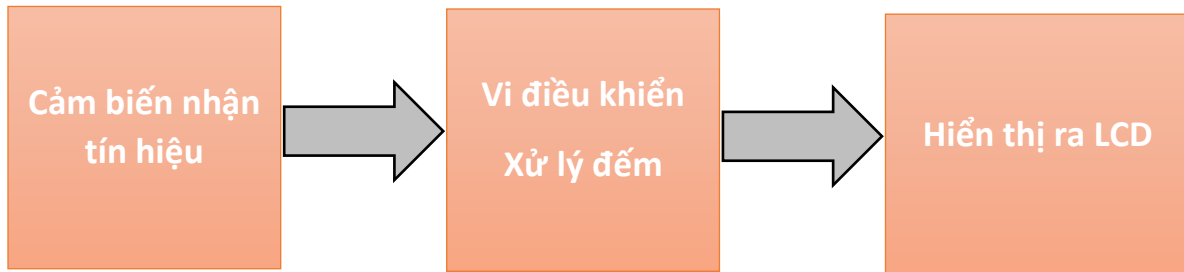
**Thông số kỹ thuật:**

- Điện áp sử dụng: 3.3~5vDC
- Nhận biết vật cản bằng ánh sáng hồng ngoại.
- Ngõ ra: Digital TTL
- Tích hợp biến trở chỉnh khoảng cách nhận biết vật cản.
- Kích thước: 3.2 x 1.4cm

## CHƯƠNG 2: NGUYÊN LÝ HOẠT ĐỘNG CƠ BẢN

### 2.1 Lưu đồ hoạt động và nguyên lý:

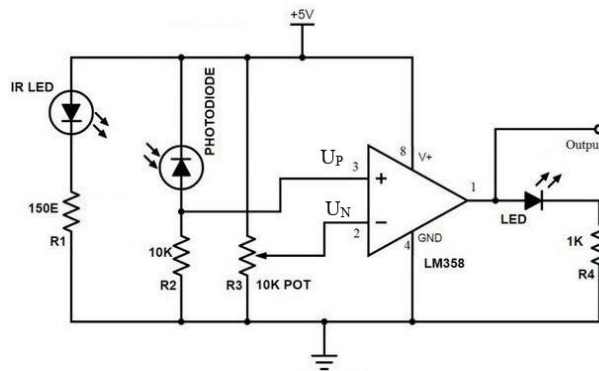
Sơ đồ khối:



### 2.2 Sơ đồ nguyên lý:

Khi cảm biến hồng ngoại nhận được tín hiệu có vật cản sẽ gửi tín hiệu đến pic. Nếu cảm biến 1 nhận được tín hiệu trước rồi mới đến cảm biến 2 thì vi điều khiển sẽ đếm lên với số người “vào”, ngược lại nếu cảm biến 2 nhận được tín hiệu rồi tới cảm biến 1 thì vi điều khiển sẽ đếm lên số người “ra”. Và led sẽ sáng khi số người “vào” khác với số người “ra”, nghĩa là trong phòng có người. Rồi từ đó xuất ra màn hình LCD.

### 2.3 Nguyên lý hoạt động của cảm biến hồng ngoại:



Hình 2.0. 5 Sơ đồ nguyên lý mạch cảm biến hồng ngoại



Khi module không có vật cản phía trước tức là không có bức xạ hồng ngoại IR hướng tới led thu hồng ngoại (đốt quang) thì đầu ra của bộ so sánh là thấp (mức 0).

Khi có vật cản phía trước (không phải màu đen), những chùm tia hồng ngoại từ LED phát đập vào vật cản và phản xạ lại LED thu làm thay đổi giá trị điện áp ở đầu vào không đảo (chân 3) của IC LM358. Khi đó, điện áp đầu vào không đảo (chân 3) được so sánh với giá trị điện áp không đổi (chân 2) ghim trên biến trở RV1, nếu giá trị điện áp ở chân 3 lớn hơn điện áp ở chân 2 (có vật cản) thì LM358 xuất ra mức 1 ( +VCC )

Led thu hồng ngoại được kết nối phân cực ngược, đầu đảo của LM358 (PIN 2) được kết nối với biến trở R3, để điều chỉnh độ nhạy của cảm biến. Đầu không đảo (PIN 3) được kết nối với đường giao nhau của đốt quang và điện trở. Điện trở R1 ,R2 được sử dụng để đảm bảo dòng điện đi qua các thiết bị LED phát (IR LED), LED thu (Photodiode). Biến trở RV1 dùng để chỉnh độ nhạy của mạch.

Điện áp ở đầu đảo hay còn được gọi là điện áp ngưỡng , có thể được đặt bằng cách xoay núm của biến trở. Điện áp cao hơn ở đầu đảo (-), cảm biến kém nhạy hơn và hạ điện áp ở đầu đảo (-), cảm biến nhạy hơn.

## CHƯƠNG 3: Chương Trình Phần Mềm Sử Dụng Hệ Thông

### 3.1 Sơ lược về phần mềm sử dụng:



Hình 16: Phần mềm lập trình MPLAB

MPLAB X IDE là phần mềm chạy trên máy tính nhằm phát triển các ứng dụng cho các bộ vi điều khiển và các bộ điều khiển tín hiệu số của Microchip. Các phần mềm có thể được sử dụng với hệ điều hành Windows, Mac và Linux. Nó được gọi là môi trường phát triển tích hợp (Integrated Development Environment) vì nó cung cấp các phương tiện toàn diện cho các nhà phát triển. Không giống như các phiên bản trước của MPLAB, MPLAB X IDE dựa trên mã nguồn mở NetBeans IDE của Oracle.

### ***3.2 Code:***

#### ***3.2.1. Code trên pic18F4550:***

```

/*
 * File:  newmain.c
 *
 * Created on July 11, 2022, 8:32 PM
 */

#pragma config PLLDIV = 1      // PLL Prescaler Selection bits (No prescale (4 MHz oscillator input
drives PLL directly))

#pragma config CPUDIV = OSC1_PLL2 // System Clock Postscaler Selection bits ([Primary Oscillator
Src: /1][96 MHz PLL Src: /2])

#pragma config USBDIV = 1      // USB Clock Selection bit (used in Full-Speed USB mode only;
UCFG:FSEN = 1) (USB clock source comes directly from the primary oscillator block with no postscale)

#pragma config FOSC = HS      // Oscillator Selection bits (EC oscillator, CLKO function on RA6 (EC))

#pragma config FCMEN = OFF    // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor
disabled)

#pragma config IESO = OFF     // Internal/External Oscillator Switchover bit (Oscillator Switchover
mode disabled)

#pragma config PWRT = OFF     // Power-up Timer Enable bit (PWRT disabled)

#pragma config BOR = ON      // Brown-out Reset Enable bits (Brown-out Reset enabled in hardware
only (SBOREN is disabled))

#pragma config BORV = 3      // Brown-out Reset Voltage bits (Minimum setting 2.05V)

#pragma config VREGEN = OFF   // USB Voltage Regulator Enable bit (USB voltage regulator
disabled)

#pragma config WDT = ON      // Watchdog Timer Enable bit (WDT enabled)

#pragma config WDTPS = 32768 // Watchdog Timer Postscale Select bits (1:32768)

```

```

#pragma config CCP2MX = OFF    // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)

#pragma config PBADEN = OFF    // PORTB A/D Enable bit (PORTB<4:0> pins are configured as analog input channels on Reset)

#pragma config LPT1OSC = OFF    // Low-Power Timer 1 Oscillator Enable bit (Timer1 configured for higher power operation)

#pragma config MCLRE = ON      // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)


#pragma config STVREN = ON     // Stack Full/Underflow Reset Enable bit (Stack full/underflow will cause Reset)

#pragma config LVP = ON       // Single-Supply ICSP Enable bit (Single-Supply ICSP enabled)

#pragma config ICPRT = OFF     // Dedicated In-Circuit Debug/Programming Port (ICPORT) Enable bit (ICPORT disabled)

#pragma config XINST = OFF     // Extended Instruction Set Enable bit (Instruction set extension and Indexed Addressing mode disabled (Legacy mode))


#pragma config CP0 = OFF      // Code Protection bit (Block 0 (000800-001FFFh) is not code-protected)

#pragma config CP1 = OFF      // Code Protection bit (Block 1 (002000-003FFFh) is not code-protected)

#pragma config CP2 = OFF      // Code Protection bit (Block 2 (004000-005FFFh) is not code-protected)

#pragma config CP3 = OFF      // Code Protection bit (Block 3 (006000-007FFFh) is not code-protected)


#pragma config CPB = OFF      // Boot Block Code Protection bit (Boot block (000000-0007FFFh) is not code-protected)

#pragma config CPD = OFF      // Data EEPROM Code Protection bit (Data EEPROM is not code-protected)


#pragma config WRT0 = OFF      // Write Protection bit (Block 0 (000800-001FFFh) is not write-protected)

```

```

#pragma config WRT1 = OFF    // Write Protection bit (Block 1 (002000-003FFFh) is not write-
protected)

#pragma config WRT2 = OFF    // Write Protection bit (Block 2 (004000-005FFFh) is not write-
protected)

#pragma config WRT3 = OFF    // Write Protection bit (Block 3 (006000-007FFFh) is not write-
protected)


#pragma config WRTC = OFF    // Configuration Register Write Protection bit (Configuration registers
(300000-3000FFFh) are not write-protected)

#pragma config WRTB = OFF    // Boot Block Write Protection bit (Boot block (000000-0007FFFh) is
not write-protected)

#pragma config WRTD = OFF    // Data EEPROM Write Protection bit (Data EEPROM is not write-
protected)


#pragma config EBTR0 = OFF    // Table Read Protection bit (Block 0 (000800-001FFFh) is not
protected from table reads executed in other blocks)

#pragma config EBTR1 = OFF    // Table Read Protection bit (Block 1 (002000-003FFFh) is not
protected from table reads executed in other blocks)

#pragma config EBTR2 = OFF    // Table Read Protection bit (Block 2 (004000-005FFFh) is not
protected from table reads executed in other blocks)

#pragma config EBTR3 = OFF    // Table Read Protection bit (Block 3 (006000-007FFFh) is not
protected from table reads executed in other blocks)


#pragma config EBTRB = OFF


#include<pic18f4550.h>


#define RS LATB4 /*PIN 4 of PORTB is assigned for register select Pin of LCD*/
#define EN LATB2 /*PIN 2 of PORTB is assigned for enable Pin of LCD */
#define ldata LATD /*PORTB(PB4-PB7) is assigned for LCD Data Output*/

```

```

#define LCD_Port TRISD /*define macros for PORTD Direction Register*/

#define _XTAL_FREQ 8000000

void LCD_Init();

void LCD_Command(char );

void LCD_Char(char x);

void LCD_String(const char *);

void LCD_String_xy(char ,char ,const char*);

void MSdelay(unsigned int);

void MSdelay(unsigned int val)
{
    unsigned int i,j;

    for(i=0;i<val;i++)
        for(j=0;j<165;j++); /*This count Provide delay of 1 ms for 8MHz Frequency */
}

void External_Interrupt_Init()
{
    TRISBbits.TRISB0=1; /* Make INT0 pin as an input pin*/
    TRISBbits.TRISB1=1; /* Make INT0 pin as an input pin*/
    INTCON2bits.INTEDG0=0;
    INTCON2bits.INTEDG1=0;
    INTCONbits.INT0IF=0; /* Clear INT0IF flag*/
    INTCONbits.INT0IE=1; /* Enable INT0 external interrupt*/
    INTCONbits.GIE=1; /* Enable Global Interrupt*/
}

```

```

    INTCONbits.PEIE=1;

    INT1IE=1;

    INT1IF=0;
}

void LCD_Init()
{
    MSdelay(15);      /* 15ms, 16x2 LCD Power on delay */

    LCD_Port = 0x00;   /* Set PORTD as output PORT for LCD data(D0-D7) pins */

    TRISBbits.TRISB4=0;

    TRISBbits.TRISB2=0;

    LCD_Command(0x38); /* uses 2 line and initialize 5*7 matrix of LCD */

    LCD_Command(0x01); /* clear display screen */

    LCD_Command(0x0c); /* display on cursor off */

    LCD_Command(0x06); /* increment cursor (shift cursor to right) */
}

void LCD_Clear()
{
    LCD_Command(0x01); /* clear display screen */
}

void LCD_Command(char cmd )
{
    ldata= cmd;        /* Send data to PORT as a command for LCD */

    RS = 0;           /* Command Register is selected */
}

```

```

    EN = 1;          /* High-to-Low pulse on Enable pin to latch data */

    EN = 0;

    MSdelay(3);
}

void LCD_Char(char dat)
{
    ldata= dat;      /* Send data to LCD */

    RS = 1;          /* Data Register is selected */

    EN=1;            /* High-to-Low pulse on Enable pin to latch data */

    EN=0;

    MSdelay(1);
}

void LCD_String(const char *msg)
{
    while((*msg)!=0)
    {
        LCD_Char(*msg);

        msg++;
    }
}

void LCD_String_xy(char row,char pos,const char *msg)

```



```

{
    char location=0;
    if(row<=1)
    {
        location=(0x80) | ((pos) & 0x0f); /*Print message on 1st row and desired location*/
        LCD_Command(location);
    }
    else
    {
        location=(0xC0) | ((pos) & 0x0f); /*Print message on 2nd row and desired location*/
        LCD_Command(location);
    }
    LCD_String(msg);
}

unsigned int vao=0,ra=0,t=0;int x=0;
char vao1[2];
char ra1[2];

void main(void) {
    OSCCON=0x72;          /* Set internal oscillator to 8MHz*/
    External_Interrupt_Init();
    LCD_Init();           /*Initialize LCD to 5*8 matrix in 4-bit mode*/
    TRISAbits.RA0=0;
    while(1)
    {

```

```

    if(vao-ra!=0)
    {
        LATAbits.LA0=1;
        LCD_String_xy(1,3,"wellcome!!");
    }

    else

    {
        LATAbits.LA0=0;
    }

        LCD_String_xy(2,0,"Vao:");
        LCD_String_xy(2,4,vao1);
        LCD_String_xy(2,11,"Ra:");
        LCD_String_xy(2,14,ra1);
        MSdelay(1000);
    }
    return;
}

void __interrupt isr(){
    if(PORTBbits.RB0==0){
        x=x*10+1;
        INT0IF=0;
    }

    if(PORTBbits.RB1==0){
        x=x*10+2;
        INT1IF=0;
    }
}

```

```

}

if((x==12)==1)

{

    vao++;

    x=0;

        vao1[0]=vao/10+48;

        vao1[1]=vao%10+48;

        ra1[0]=ra/10+48;

        ra1[1]=ra%10+48;

}

if((x==21)==1)

{

    ra++;

    x=0;

        vao1[0]=vao/10+48;

        vao1[1]=vao%10+48;

        ra1[0]=ra/10+48;

        ra1[1]=ra%10+48;

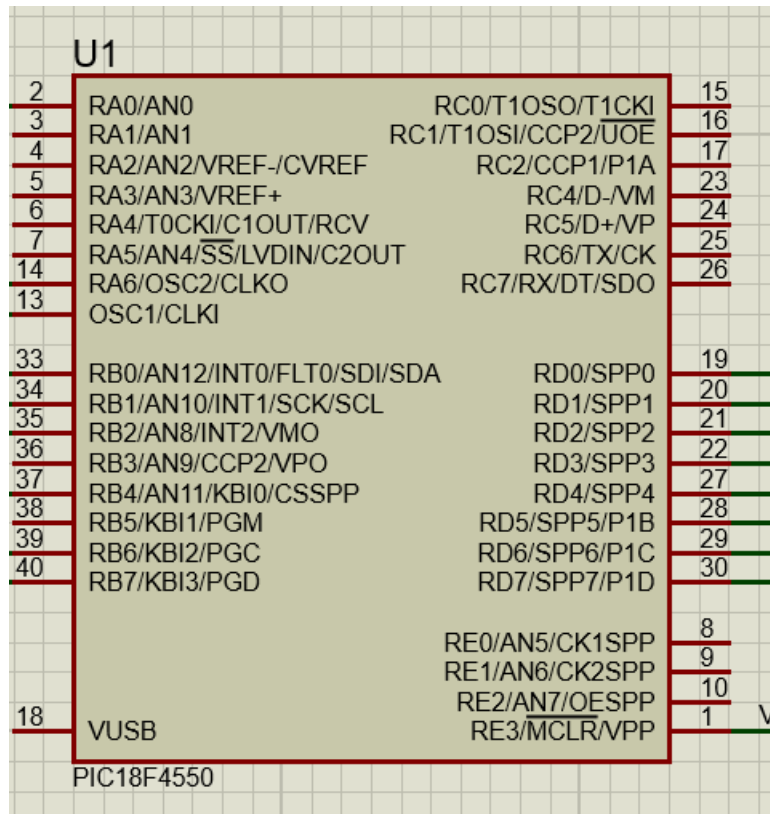
}

}

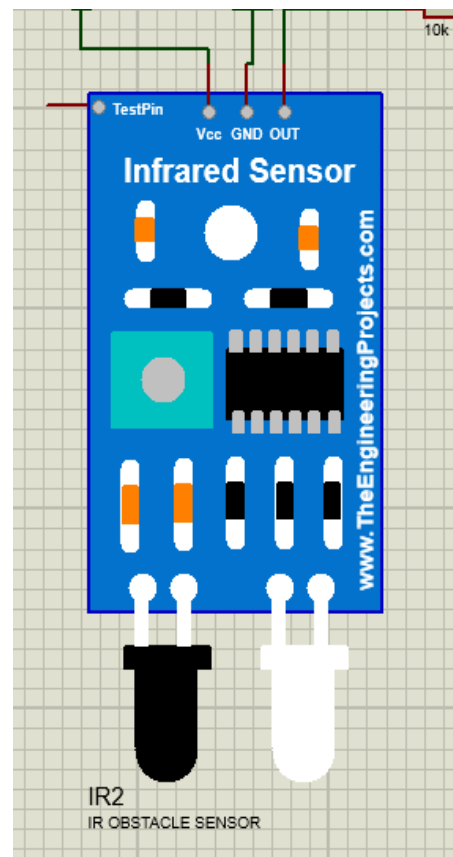
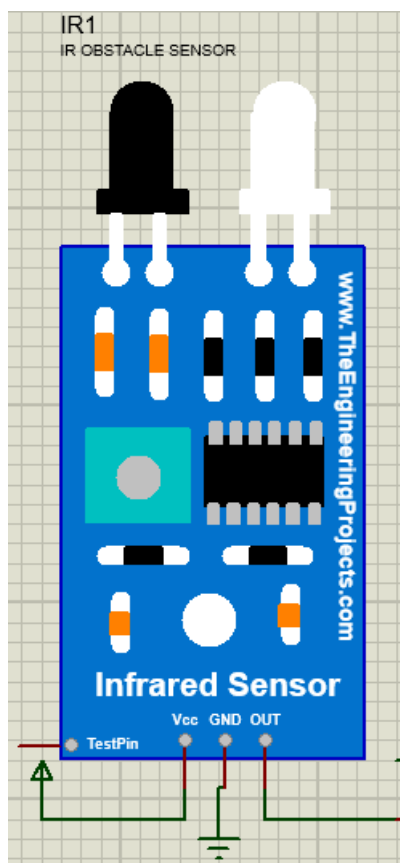
```

### ***3.3 Hình mô phỏng proteus:***

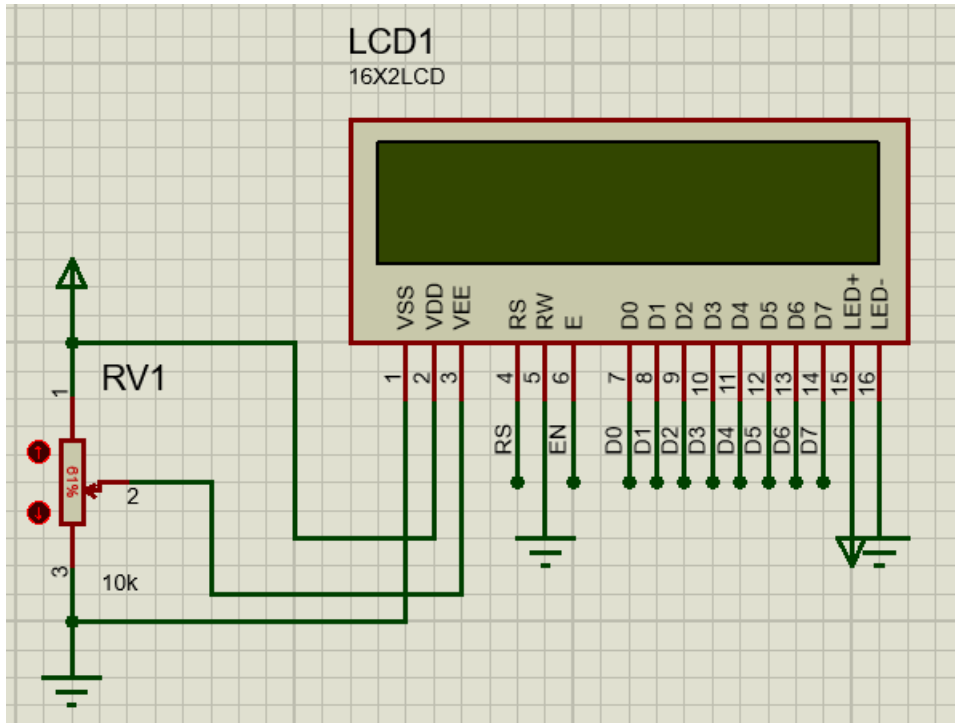
***Pic18f4550:***



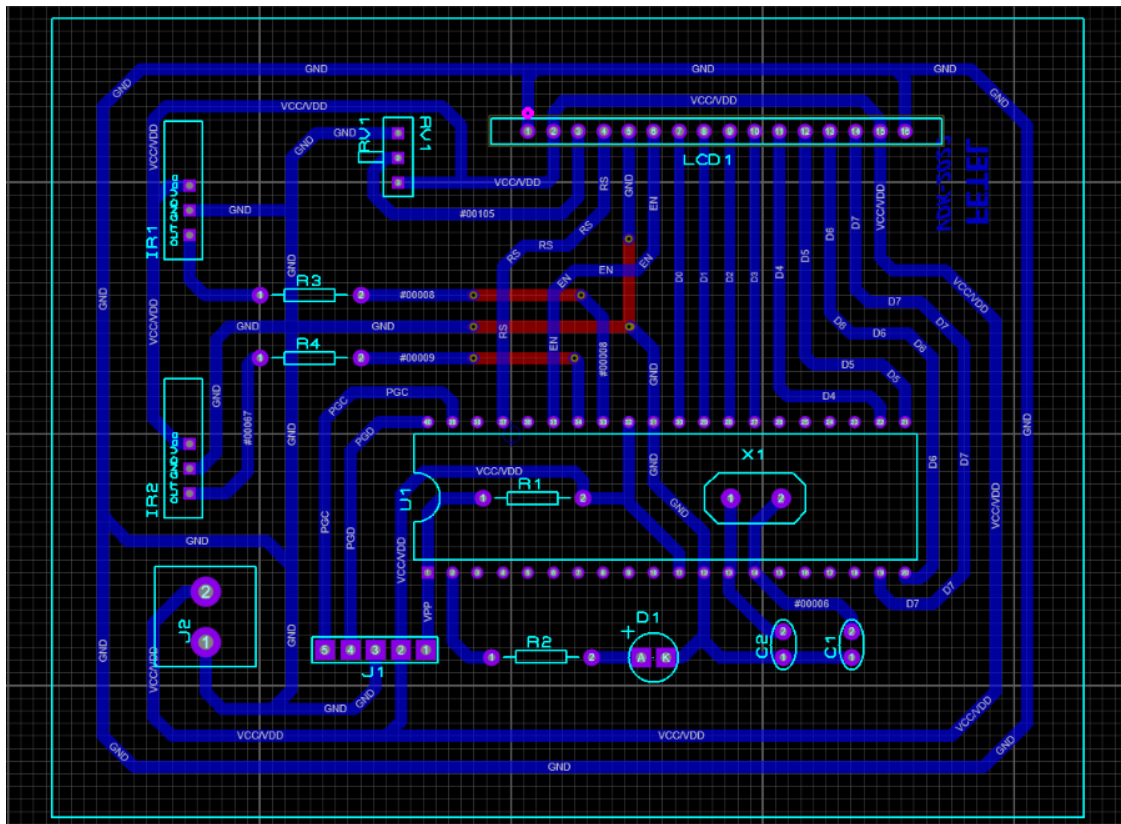
*Cảm biến hồng ngoại:*



*LCD:*



*Bản in vẽ mạch pcb:*



The diagram illustrates a PIC18F4550 microcontroller circuit. The PIC is connected to an LCD1602 display (LM016L) showing 'welcome!!', 'Vao:03', and 'Ra:00'. It also controls an LED (D1) and is interfaced with two potentiometers (RV1 and RV2) via resistors R2 and R3. The PIC's pinout is shown on the right, with various peripheral modules like RC0/TIO, RD0/SPR, and RE0/AN listed.

## KẾT LUẬN

Do thời gian vừa thi xong và trình độ có hạn nên nhóm em không tránh khỏi những sai sót. Vì vậy mong thầy góp ý xây dựng giúp đỡ để hoàn thành đề tài . Việc áp dụng vi điều khiển vào công nghệ điện tử hiện đại đã cho ra đời các sản phẩm tiện ích thông minh, tăng khả năng làm việc, tính toán của các mạch điện tử và các cảm biến. Ngoài ra với giá thành rẻ, đơn giản, dễ sử dụng nên các sản phẩm này được ứng dụng rộng rãi trên nhiều lĩnh vực. Đối với các trường học, cơ quan, doanh nghiệp sản xuất, các nhà máy hay các siêu thị, cửa hàng ... thì việc kiểm soát số lượng học sinh, nhân viên, sản phẩm... cần được quản lý một cách khoa học và tự động. Để đáp ứng nhu cầu đó, nhóm em đã thiết kế và chế tạo một hệ thống đếm số lượng học sinh tự động góp phần giảm thời gian kiểm soát đồng thời vẫn đảm bảo độ chính xác cao. Nhóm em đã cố gắng hoàn thiện đề tài một cách tốt nhất. Nhưng với những kiến thức và sự hiểu biết còn hạn chế nên chắc chắn sẽ không tránh khỏi những thiếu sót. Nhóm em mong các thầy cô cùng các bạn đóng góp ý kiến để đề tài của nhóm em có thể hoàn thiện hơn.

### ***PHIẾU ĐÁNH GIÁ ĐỒ ÁN VI ĐIỀU KHIỂN***

Tiêu chí			Điểm	Tổng điểm tối đa	Đánh giá
Hệ thống Hoạt động. Có mạch PCB	PCB	Đẹp không nối dây	2	8	
		Có nối dây	1		
	Làm mô hình Demo		1		
	Sử dụng ngắt, timer, uart, adc, ...		1		
	Hiểu	Trả lời được các câu hỏi	1		
		Giải thích nguyên lý hoạt động	1		
	Trình bày	Cuốn báo cáo trình bày đầy đủ nội dung	0.5		
		Slide trình bày dễ hiểu	0.5		
	Demo hoạt động tốt		1		
Hệ thống hoạt động. Không có mạch PCB	Demo hoạt động tốt		1	5	
	Sử dụng ngắt, timer, uart, adc, ...		1		
	Hiểu	Trả lời được các câu hỏi	1		
		Giải thích nguyên lý hoạt động	1		
	Trình bày	Cuốn báo cáo trình bày đầy đủ nội dung	0.5		
		Slide trình bày dễ hiểu	0.5		

Điểm tổng kết: .....