

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH**  
**KHOA CÔNG NGHỆ ĐIỆN TỬ**



STT	Họ và tên	MSSV
1	Nguyễn Văn Đức Tuấn	21069361
2	Hồ Việt Dương Tài	21052521
3	Phạm Thanh Lịch	21062441

**ĐỀ TÀI:**  
**ĐIỀU KHIỂN CỬA NHÀ**

Chuyên ngành: Công nghệ kỹ thuật máy tính  
Mã chuyên ngành: DHDTMT17B

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

## THÔNG TIN CHUNG

Họ và tên sinh viên: Nguyễn Văn Đức Tuân ..... MSSV: 21069361 .....

Lớp : 420300353502 ..... Khóa: K17 .....

Chuyên ngành : Công nghệ kỹ thuật điện tử ..... Mã chuyên ngành: 7480108.....

SĐT : 0383943756 .....

Email : tuangato147@gmail.com.....

Địa chỉ liên hệ : 118/11 đường số 8, phường 11, Gò Vấp, Hồ Chí Minh  
.....

Tên đề tài : Điều khiển cửa nhà .....

.....

.....

Người hướng dẫn : Ths Nguyễn Duy Khanh .....

.....

.....

SĐT : .....

Email : .....

Cơ quan công tác : .....

*Tp. Hồ Chí Minh, ngày tháng năm 20.....*

**Người hướng dẫn**  
(Ký và ghi rõ họ tên)

**Sinh viên**  
(Ký và ghi rõ họ tên)

## THÔNG TIN CHUNG

Họ và tên sinh viên: Phạm Thanh Lịch..... MSSV: 21062441 .....

Lớp : 420300353502 ..... Khóa:K17 .....

Chuyên ngành : Công nghệ kỹ thuật điện tử ..... Mã chuyên ngành:7480108.....

SĐT :0914861912 .....

Email :lichpham1125@gmail.com.....

Địa chỉ liên hệ :14b1 Hà huy giáp, Thạnh lộc ,Quận  
12 .....

Tên đề tài : Điều khiển cửa nhà .....

.....

.....

Người hướng dẫn : Ths Nguyễn Duy Khanh.....

.....

.....

SĐT : .....

Email : .....

Cơ quan công tác : .....

*Tp. Hồ Chí Minh, ngày tháng năm 20.....*

**Người hướng dẫn**  
(Ký và ghi rõ họ tên)

**Sinh viên**  
(Ký và ghi rõ họ tên)

## THÔNG TIN CHUNG

Họ và tên sinh viên: Hồ Viết Dương Tài..... MSSV: 21052521 .....

Lớp : 420300353502 ..... Khóa:K17 .....

Chuyên ngành : Công nghệ kỹ thuật điện tử ..... Mã chuyên ngành: 7480108.....

SĐT :  
0901553129 .....

Email : hovietduongtai@gmail.com .....

Địa chỉ liên hệ : 85/9A, Bình Thành, Bình Tân,  
TP.HCM .....

Tên đề tài : Điều khiển cửa nhà .....

.....

.....

Người hướng dẫn : Ths Nguyễn Duy Khanh.....

.....

.....

SĐT : .....

Email : .....

Cơ quan công tác : .....

*Tp. Hồ Chí Minh, ngày tháng năm 20.....*

**Người hướng dẫn**  
(Ký và ghi rõ họ tên)

**Sinh viên**  
(Ký và ghi rõ họ tên)

## MỤC LỤC

MỞ ĐẦU.....	1
1. Đặt vấn đề .....	1
2. Mục tiêu của đề tài .....	1
3. Đối tượng và phạm vi nghiên cứu.....	2
4. Ý nghĩa thực tiễn của đề tài.....	2
5. Giới hạn đề tài .....	2
CHƯƠNG 1    TỔNG QUAN VỀ ĐỀ TÀI.....	4
1.1    Sơ lược về điều khiển cửa bằng cảm biến ánh sáng.....	4
1.1.1    Tìm hiểu nhu cầu và phân tích mạch .....	4
1.1.2    Những sản phẩm đã được đưa ra cuộc sống .....	4
1.2    Bối cảnh hiện nay .....	6
1.3    Các tiêu chí về kĩ thuật của sản phẩm .....	6
1.3.1    Sản phẩm và những chức năng cần thiết .....	6
1.3.2    Tiêu chí kĩ thuật .....	6
CHƯƠNG 2    CƠ SỞ LÝ THUYẾT.....	8
2.1    Phần cứng .....	8
2.1.1    Giới thiệu về Arduino nano .....	8
2.1.2    Giới thiệu về cảm biến màu sắc TCS34725 .....	11
2.1.3    Giới thiệu về động cơ Servo .....	13
2.1.4    Giới thiệu về cách ly quang Opto 4N35 .....	14
2.1.5    Các chuẩn giao tiếp được sử dụng .....	15
2.2    Phần mềm .....	18
2.2.1    Giới thiệu về ứng dụng Visual Studio .....	18
2.2.2    Giới thiệu về Arduino IDE .....	19
CHƯƠNG 3    PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....	21
3.1    Thiết kế sơ đồ khối hệ thống .....	21
3.2    Sơ đồ nguyên lý .....	22
3.2.1    Tính toán thông số mạch.....	23

3.2.2	Tính toán thông số mạch.....	25
3.3	Lưu đồ giải thuật.....	27
3.4	Test bo mạch.....	28
3.4.1	Ảnh test bo mạch thực tế .....	28
3.5	Thiết kế mạch PCB trên ứng dụng Altium.....	29
3.5.1	Tính toán mạch PCB.....	29
3.5.2	Mô hình 2D trên hai lớp: .....	30
3.5.3	Mô hình 2D của lớp Top và lớp Bottom: .....	31
3.5.4	Mô hình 3D của mạch:.....	31
3.6	tính toán tốc độ truyền uart.....	32
CHƯƠNG 4	THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ.....	33
4.1	Hình ảnh thực tế đo bằng oscilloscope.....	33
4.1.1	Sóng nguồn đi vào .....	33
4.1.2	Đo tín hiệu uart của mạch bằng oscilloscope .....	35
4.2	Thiết kế giao diện C#.....	39
4.2.1	Một vài minh chứng về truyền nhận uart.....	40
4.3	Ảnh thực tế của mạch .....	42
4.3.1	Hình ảnh kết nối mạch với các thiết bị ngoại vi .....	42
4.4	Mô hình cửa.....	44
4.4.1	Ảnh mô hình mô phỏng bằng tinkercad .....	44
4.4.2	Ảnh thực tế sản phẩm mô hình .....	45
4.5	Đánh giá kết quả đạt được .....	48
KẾT LUẬN VÀ KIẾN NGHỊ.....		48
TÀI LIỆU THAM KHẢO.....		50

## MỤC LỤC HÌNH ẢNH

Hình 1: Hệ thống phân loại cà chua theo màu sắc [1] .....	5
Hình 2: Hệ thống phân loại sản phẩm theo màu sắc [2] .....	6
Hình 3: Mặt trước và mặt sau của arduino nano .....	8
Hình 4: Sơ đồ khối chi tiết của arduino nano.....	9
Hình 5: Mặt trước và sau của cảm biến màu sắc tcs 34725 .....	11
Hình 6: Sơ đồ mạch của tcs34725.....	12
Hình 7: Sơ đồ khối chi tiết của tcs34725 .....	12
Hình 8: Micro servo 9g .....	14
Hình 9: Opto 4N35 .....	15
Hình 10: Sơ đồ mạch opto4N35.....	15
Hình 11: Minh họa cách truyền uart .....	16
Hình 12: Cách truyền bit của uart .....	17
Hình 13: Hình minh họa visul studio .....	18
Hình 14: Minh họa giao diện IDE arduino .....	19
Hình 15: Sơ đồ khối hệ thống .....	21
Hình 16: Sơ đồ nguyên lý của mạch .....	22
Hình 17: Sơ đồ mạch nguồn hạ áp 12V-9V .....	22
Hình 18: Sơ đồ nguyên lý opto 4N35 .....	25
Hình 19: test board ,mạch với các thiết bị ngoại vi .....	28
Hình 20: Test truyền uart giữa cảm biến và arduino.....	28
Hình 21: Test board mạch và giao diện C#.....	29
Hình 22: Bố trí linh kiện và đi dây 2 lớp .....	30
Hình 23: Ảnh đi dây PCB của lớp top và bottom .....	31
Hình 24: Ảnh 3D của mạch PCB .....	31
Hình 25: nguồn vào oscilloscope .....	33
Hình 26: 1 chu kì của nguồn vào .....	34
Hình 27: ảnh đo thực tế.....	34
Hình 28: sóng đo truyền tín hiệu uart .....	35
Hình 29: truyền kí tự từ giao diện sang mạch bằng uart.....	35
Hình 30: độ rộng của tín hiệu TX với 10 bit.....	37
Hình 31: độ rộng của tín hiệu RX với 10 bit.....	37
Hình 32: độ dài của nhận về và truyền đi của uart(TX/RX) .....	38
Hình 33: giao diện C# .....	39
Hình 34: truyền 100 kí tự A .....	40
Hình 35: hiển thị màu đỏ từ cảm biến.....	40
Hình 36: hiển thị màu xanh lá từ cảm biến .....	41
Hình 37: hiển thị màu xanh dương từ cảm biến.....	41

Hình 38: Ảnh mạch PCB thực tế.....	42
Hình 39: Ảnh kết nối với các thiết bị ngoại vi.....	42
Hình 40: cấp nguồn và chạy thử .....	43
Hình 41:kết nối với các thiết bị ngoại vi.....	43
Hình 42: Ảnh chạy mạch.....	44
Hình 43: ảnh mặt trước của mô hình.....	44
Hình 44: ảnh sau của mô hình.....	45
Hình 45: Quá trình thi công sản phẩm .....	46
Hình 46: Mặt trước mô hình .....	47
Hình 47: Mặt bên cạnh mô hình.....	47
Hình 48: Bên trong mô hình.....	48



# MỞ ĐẦU

## 1. Đặt vấn đề

Ở những môi trường điều kiện thiếu sáng chẳng hạn như cửa nhà kho hoặc những nơi điều kiện chiếu sáng hạn chế việc đóng mở cửa có thể trở nên khó khăn và kém an toàn, ta cần một giải pháp thông minh hơn về việc chỉ phụ thuộc vào khóa vật lý cơ bản, trên thị trường giờ đã có rất nhiều loại khóa sử dụng cảm biến thông minh phù hợp với từng mục đích riêng, ta thấy một số nơi chỉ cần cơ bản và hiệu quả mà không cần đầu tư quá nhiều, cảm biến màu sắc là một ý tưởng không tệ về việc thiết kế để làm khóa cửa thông minh giải quyết các vấn đề cần thiết như bảo mật , tự động hóa cũng như sự tiện lợi và hiện tại mà nó mang lại và việc đóng mở cửa dựa trên màu sắc cụ thể có thể giải quyết vấn đề trên.

Đề tài có rất nhiều tiềm năng phát triển trong tương lai, về mặt kinh tế nó có chi phí tối ưu so với những công nghệ đắt đỏ ngoài kia, cắt giảm chi phí lắp đặt và vận hành, tiếp theo là dễ tiếp cận và sửa chữa cũng như bảo trì thiết bị, hứa hẹn rất nhiều về tiềm năng sắp tới. Sau tất cả những ý trên thì nó còn rất phù hợp cho những dự án nhỏ để có thể thực hiện xây dựng và phát triển cho sinh viên.

Với thực trạng công nghệ hóa hiện nay, việc chúng ta áp dụng công nghệ vào đời sống ngày một phổ biến hơn, đưa hệ thống tự động vào đời sống đang dần là xu thế của thế giới, tận dụng cảm biến màu sắc để đạt được những điều mong muốn nhằm tăng sự an toàn , công sức cũng như tính tự động hóa trong công việc.

## 2. Mục tiêu của đề tài

- Xây dựng được hệ thống mở cửa tự động
- Hoạt động ổn định và chính xác
- Đề tài xây dựng áp dụng được vào mô hình
- Truyền nhận dữ liệu trực quan trên giao diện C#

### **3. Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu bao gồm arduino nano là bo mạch có khả năng lập trình để xử lý dữ liệu và điều khiển các thiết bị ngoại vi có vai trò nhận dữ liệu từ cảm biến màu sắc và điều khiển đóng mở qua động cơ servo, cảm biến màu sắc TCS34725 là cảm biến được 3 màu sắc cơ bản và động cơ servo SG90 .

Phạm vi nghiên cứu trong các điều kiện thiếu sáng khác nhau và ngưỡng màu mà hệ thống có thể hoạt động 1 cách hiệu suất nhất, hoạt động tốt nhất vào khoảng không gian gần, kích thước nhỏ nhất có thể để tránh chiếm diện tích tiếp xúc, và chủ thể chịu tác động chính đó là cửa đóng mở.

### **4. Ý nghĩa thực tiễn của đề tài**

Dự án cửa đóng mở tự động dựa trên màu sắc thông qua cảm biến có sự thực tiễn lớn trong việc cải thiện và giải quyết vấn đề an toàn, bảo mật và hiệu quả trong những môi trường có ánh sáng yếu, bằng cách sử dụng cảm biến để tự động đóng mở giúp giảm thiểu rủi ro hơn, đồng thời nó cũng được tối ưu hóa cho những nơi có điều kiện như vậy chẳng hạn như là nhà kho hoặc những nơi có ánh sáng không ổn định, tăng tính bảo mật và trực quan khi dữ liệu trạng thái được cập nhật qua giao diện c# từ đó nâng cao hiệu quả công việc và an toàn cho người dùng.

Đề tài giải quyết vấn đề cho những người cần 1 giải pháp thông minh và tự động hóa cho cánh cửa của mình, những nơi có phòng quan trọng hoặc là riêng tư.

Đề tài có tính hiệu quả khá khi áp dụng những cảm biến và công nghệ vào đồ vật, tính hiệu quả khá vì cảm biến ánh sáng còn bị giới hạn bởi nhiều mặt như điều kiện môi trường , giới hạn phần cứng...

### **5. Giới hạn đề tài**

Có một vài yếu tố ảnh hưởng đến đề tài như kinh tế ,kiến thức, chuyên môn, kinh nghiệm, nên ta sẽ khoanh vùng đề tài trong những yêu cầu sau đây:

- Hệ thống sẽ được theo dõi và cập nhật trực quan trên laptop thông qua dây cáp
- Kích thước mô hình nhỏ gọn
- Tận dụng những thư viện đã được xây dựng sẵn để thực hiện dự án
- Giao diện được thiết kế bằng C#, đơn giản và có những chức năng cơ bản cần thiết

## **CHƯƠNG 1      TỔNG QUAN VỀ ĐỀ TÀI**

### **1.1   Sơ lược về điều khiển cửa bằng cảm biến ánh sáng**

#### ***1.1.1   Tìm hiểu nhu cầu và phân tích mạch***

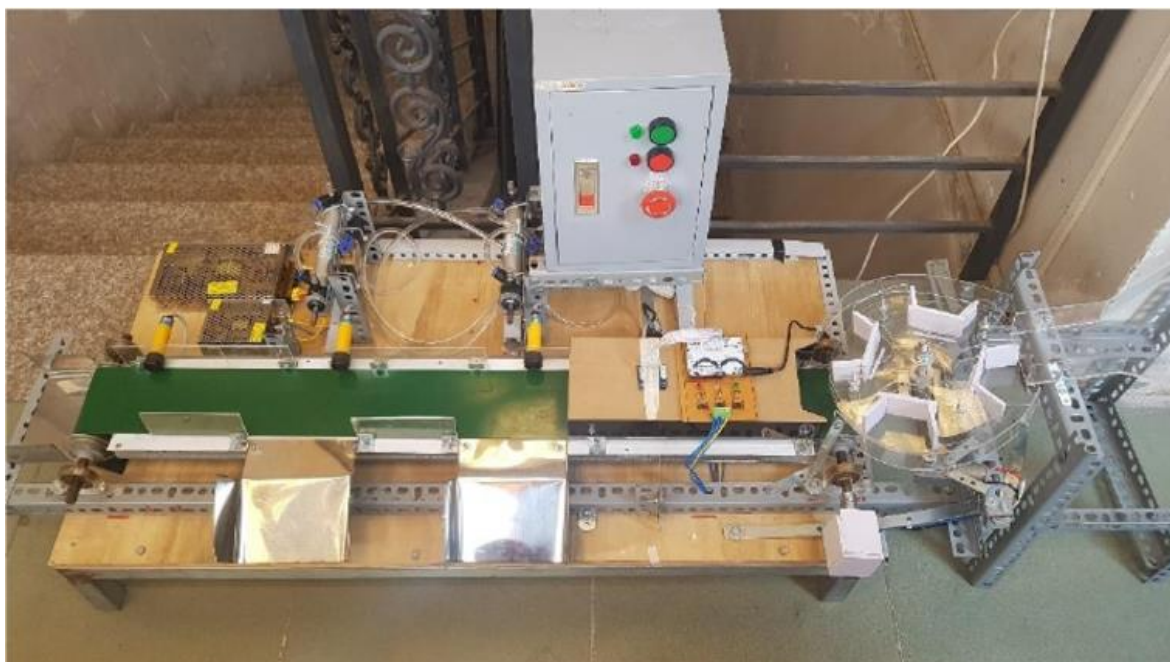
Cảm biến nhận diện ngày càng được phát triển qua từng ngày. Từ những biến thô sơ cơ bản nhất đến những cảm biến nhận diện hiện đại nhất, mỗi cảm biến sinh ra đều có mục đích và nhu cầu riêng. Cảm biến màu sắc được ra đời cũng nhằm mục đích là nhận diện, đúng như cái tên của cảm biến, nó sinh ra là để nhận diện màu sắc cơ bản nhất đó là RGB ( red, green, blue ). Cùng với mạch giao tiếp I2C hay UART nó có thể truyền nhận được rất nhiều vi xử lí, và từ đó nhiều bài toán khó khăn cũng đã được giải quyết từ đây, ở đây, chúng ta sẽ tận dụng nó để xử lí vấn đề mở cửa trong các điều kiện khó khăn cũng như mang sự tự động hóa và trong đời sống con người.

#### ***1.1.2   Những sản phẩm đã được đưa ra cuộc sống***

Trong cuộc sống gần đây, ta đã có được rất nhiều hệ thống tự động dùng cảm biến màu sắc để phục vụ cho từng mục đích riêng, sau đây là một vài sản phẩm về cảm biến.

##### ***1.1.2.1   Hệ thống phân loại cà chua theo màu sắc***

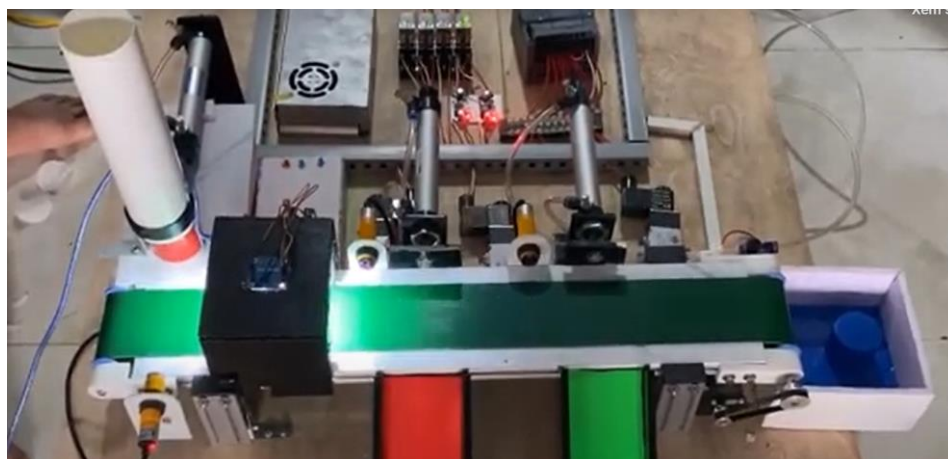
- Sử dụng điện áp: 24VDC
- Có đèn báo và nút bấm
- Sử dụng băng tải để chuyển tải đồ
- Dùng hệ thống đẩy khí nén



Hình 1: Hệ thống phân loại cà chua theo màu sắc [1]

#### *1.1.2.2 Hệ thống phân loại sản phẩm theo màu sắc*

- Sử dụng nguồn: 24VDC
- Bộ truyền dẫn động cơ, động cơ DC
- Cơ cấu đẩy
- Relay trung gian



Hình 2: Hệ thống phân loại sản phẩm theo màu sắc [2]

## 1.2 Bối cảnh hiện nay

Trên thị trường đã có rất nhiều sản phẩm làm về phân biệt màu sắc, nhưng chủ yếu là sẽ phân biệt các đồ vật trên băng truyền là phổ biến, chưa có đề tài nào làm về cửa tự động làm về màu sắc.

## 1.3 Các tiêu chí về kĩ thuật của sản phẩm

### 1.3.1 Sản phẩm và những chức năng cần thiết

Sản phẩm sẽ có bóng led trắng để phản xạ màu về cảm biến để nhận diện màu sắc, nhận diện được riêng từng màu sắc rõ ràng để có thể thực hiện được yêu cầu mong muốn được đề ra. Ta sẽ có 2 động cơ servo , 1 để điều khiển cửa và 1 ta sẽ làm chốt khóa, có chức năng mở khóa cửa và mở cửa tự động. trạng thái màu sắc sẽ được cập nhật trên giao diện.

### 1.3.2 Tiêu chí kĩ thuật

- Hoạt động ổn định trong khoảng từ 5 đến 9 v
- Nhận diện màu chính xác và phản hồi ngay lập tức

- Hoạt động độ trễ thấp không bất ổn định
- Mạch không cầu kì

### **Tín hiệu đầu vào**

Đầu vào của tín hiệu là ánh sáng trắng được phản xạ lại, chính xác hơn là ánh sáng trắng từ đèn của cảm biến tới vật thể mang màu sắc và phản xạ lại về cảm biến

### **Tín hiệu đầu ra**

- Tín hiệu đầu ra của cảm biến là tín hiệu điện được khuếch đại và xử lý rồi truyền thông qua I2C hoặc UART.
- Tín hiệu điều khiển được phát ra dưới tín hiệu dạng sóng xung vuông được điều khiển bởi arduino nano truyền tới các servo

## CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

### 2.1 Phần cứng

#### 2.1.1 Giới thiệu về Arduino nano

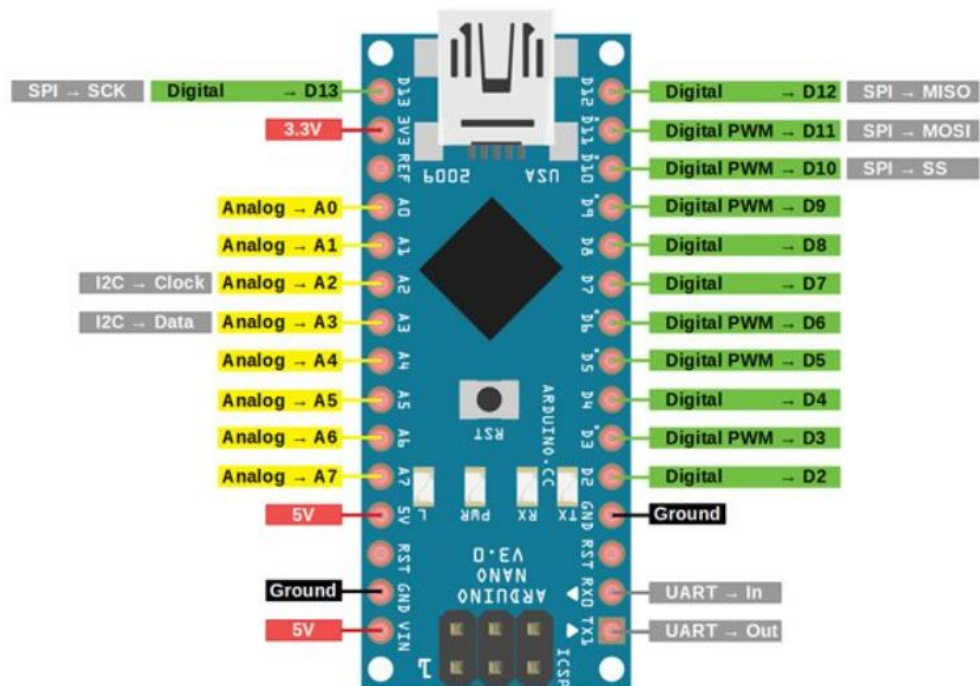
Arduino Nano là một bảng mạch điện tử có kích thước nhỏ gọn bằng 1 nửa đồng xu gấp lại, được thiết kế dựa trên ATmega328P ra mắt vào năm 2008 và tương đối thân thiện với breadboard. Arduino Nano cung cấp các tính năng và thông số kỹ thuật giống với bảng điện tử Arduino Uno tuy nhiên với kích thước nhỏ gọn hơn rất nhiều.

Arduino Nano có ưu điểm là lựa chọn được công suất phù hợp nhất với điện áp của nó, có thể lập trình trực tiếp từ máy vi tính rất thuận tiện và dễ dàng. Đặc biệt, Arduino Nano pinout có thiết kế rất gọn nhẹ, kích thước 185 mm x 430 mm với trọng lượng khoảng 7g. Nhờ ưu điểm trên mà Arduino Nano được sử dụng rất rộng rãi trong cuộc sống hiện đại ngày nay.



Hình 3: Mặt trước và mặt sau của arduino nano





Hình 4: Sơ đồ khối chi tiết của arduino nano

Chip xử lý	Atmel ATmega168 or ATmega328
Điện áp hoạt động (mức logic cao)	5 V
Điện áp vào (khuyến nghị)	7-12 V
Điện áp vào (giới hạn)	6-20 V
Số chân I/O digital	14 (6 chân đầu ra PWM)
Số chân đầu vào analog	8
Dòng Dc trên chân I/O	40 mA
Flash Memory	16 KB (ATmega168) hoặc 32 KB (ATmega328) với 2 KB sử dụng cho bootloader
SRAM	1 KB (ATmega168), 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168), 1 KB (ATmega328)
Tốc độ xung	16 MHz
Kích thước	0.73" x 1.70"

## **Nguồn cấp**

**Nguồn USB:** Cổng mini USB trên Arduino Nano có thể được sử dụng để cấp nguồn và lập trình. Nguồn USB có điện áp 5V.

**Nguồn ngoài (Vin):** Chân Vin có thể nhận điện áp từ 7V đến 12V. Điện áp này sẽ được điều chỉnh xuống 5V nhờ bộ điều chỉnh áp tích hợp (voltage regulator). Khi sử dụng nguồn ngoài, điện áp khuyên dùng là từ 7V đến 9V để tránh làm nóng chip điều chỉnh áp.

Các chân nguồn khác:

**5V:** Cung cấp nguồn 5V từ bộ điều chỉnh áp hoặc từ nguồn USB.

**3.3V:** Cung cấp nguồn 3.3V, nhưng dòng tối đa chỉ khoảng 50 mA.

**GND (Ground):** Chân nối đất, dùng làm điểm tham chiếu cho các tín hiệu và nguồn.

## **Đầu vào và đầu ra.**

Arduino Nano có tổng cộng 22 chân I/O, bao gồm cả chân kỹ thuật số và chân analog.

*Các chân kỹ thuật số (Digital pins):*

**D0 đến D13:** Tổng cộng có 14 chân kỹ thuật số. Các chân này có thể được cấu hình là đầu vào (input) hoặc đầu ra (output).

**PWM (Pulse Width Modulation):** Có 6 chân hỗ trợ PWM (D3, D5, D6, D9, D10, D11), giúp tạo tín hiệu xung PWM để điều khiển độ sáng LED, tốc độ động cơ, v.v.

*Các chân Analog (Analog pins):*

**A0 đến A7:** Có 8 chân analog, mỗi chân có độ phân giải 10-bit (tương ứng với giá trị từ 0 đến 1023). Các chân này được sử dụng để đọc tín hiệu analog từ các cảm biến (như cảm biến nhiệt độ, độ ẩm, v.v.).

*Các chân giao tiếp đặc biệt:*

**UART (Serial):** D0 (RX) và D1 (TX), dùng để giao tiếp nối tiếp (serial) với máy tính hoặc các thiết bị khác.

**I2C (Inter-Integrated Circuit):** A4 (SDA) và A5 (SCL), dùng để giao tiếp với các thiết bị I2C như cảm biến màu TCS34725.

**SPI (Serial Peripheral Interface):** D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK), dùng để giao tiếp tốc độ cao với các module như SD card, màn hình LCD, v.v.

**Reset:** Có chân RESET để khởi động lại Arduino Nano. Chân này có thể được kéo xuống mức thấp (LOW) để reset.

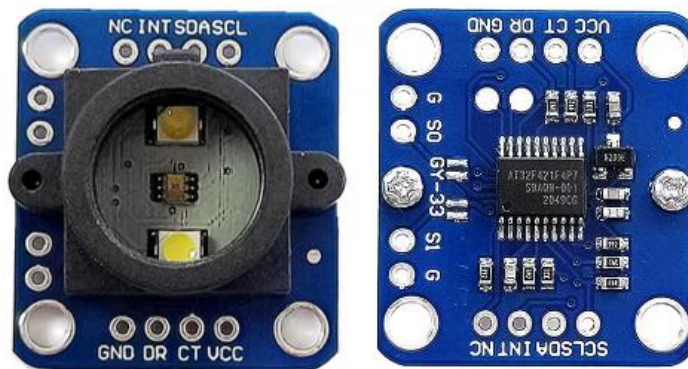
## **Bộ nhớ**

Atmega168 có bộ nhớ flash 16 KB để lưu giữ code (trong đó có 2 KB được sử dụng cho bootloader); ATmega328 có 32 KB, (cũng với 2 KB được sử dụng cho

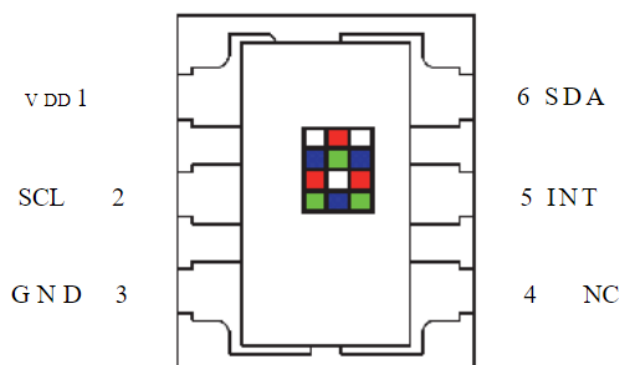
bootloader). Atmega168 có 1 KB SRAM và 512 byte EEPROM (có thể được đọc và ghi bởi thư viện EEPROM); ATmega328 có 2 KB SRAM và 1 KB EEPROM.

### **2.1.2 Giới thiệu về cảm biến màu sắc TCS34725**

Cảm biến màu RGB TCS34725 giao tiếp I2C và UART là phiên bản cải tiến của cảm biến màu TCS3200, được sử dụng để nhận dạng màu sắc bằng cách đo phản xạ 3 màu sắc cơ bản trên vật thể là đỏ, xanh lục và xanh dương sau đó phân tích và truyền thông số đo được của các màu này thông qua giao tiếp I2C hoặc UART. Cảm biến màu sắc TCS34725 giao tiếp I2C và UART có tích hợp MCU trên mạch để chuyển đổi từ giao tiếp I2C của TCS34725 sang giao tiếp UART hoặc I2C của MCU giúp dễ dàng giao tiếp và lập trình, ta có thể dễ dàng lựa chọn giao tiếp muốn sử dụng bằng cách Set các Jumper trên mạch.



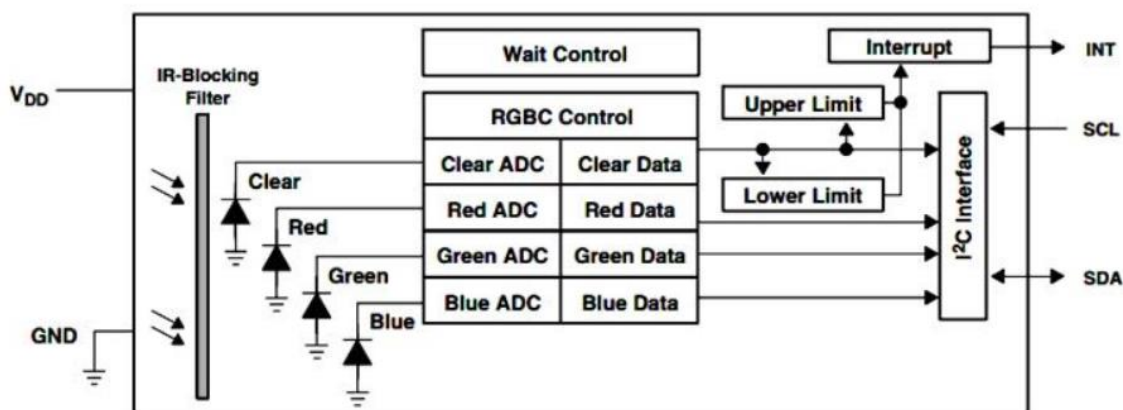
Hình 5: Mặt trước và sau của cảm biến màu sắc tcs 34725



Hình 6: Sơ đồ mạch của tcs34725

1	V DD	Chân cấp nguồn điện dương
2	SCL	Đầu vào xung clock I2C
3	GND	Chân nối đất IC
4	NC	Không kết nối
5	INT	Đầu ra ngắt, mở công
6	I / O	Đầu ra dữ liệu nối tiếp I2C

Sơ đồ bên trong của TCS34725 được hiện thị dưới đây:



Hình 7: Sơ đồ khối chi tiết của tcs34725

**Thông số kỹ thuật:**

IC chính: TCS34725

Điện áp sử dụng: 3~5VDC

Dòng tiêu thụ: 15mA

Giải màu sắc đo: RGB 0~255

Tích hợp MCU xử lý và chuyển đổi giao tiếp.

Giao tiếp có thể lựa chọn qua Jumper:

- I2C của TCS34725.
- I2C của MCU.
- UART của MCU.

Kích thước: 24.3 x 26.7mm

**2.1.3 Giới thiệu về động cơ Servo**

Servo là một loại động cơ điện đặc biệt, khác biệt so với động cơ thông thường ở chỗ nó chỉ hoạt động khi nhận được tín hiệu điều khiển. Thay vì quay liên tục khi cấp nguồn, động cơ servo sẽ quay theo một góc cụ thể từ 0° đến 180° dựa trên các xung điều khiển PPM (Pulse Position Modulation). Tùy theo ứng dụng, các loại servo có kích thước, trọng lượng và cấu trúc khác nhau.

**Nguyên lý hoạt động:**

Servo SG90 hoạt động dựa trên tín hiệu PWM (Pulse Width Modulation) để xác định góc xoay của trục.

Tín hiệu điều khiển:

Servo nhận tín hiệu PWM từ một vi điều khiển (như Arduino).

Chu kỳ tín hiệu PWM của Servo SG90 là 20ms (50Hz).

Độ rộng của xung PWM (thường từ 1ms đến 2ms) quyết định góc xoay của servo:

- 1ms: Góc 0° (vị trí trái nhất).
- 1.5ms: Góc 90° (vị trí trung tâm).
- 2ms: Góc 180° (vị trí phải nhất).



Hình 8: Micro servo 9g

### **Thông Số Kỹ Thuật Động Cơ Servo SG90**

Khối lượng: 9g

Kích thước: 23mmX12.2mmX29mm

Momen xoắn: 1.8kg/cm

Tốc độ hoạt động: 60 độ trong 0.1 giây

Điện áp hoạt động: 4.8V(~5V)

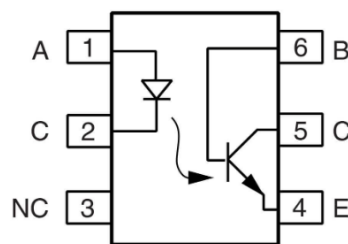
Nhiệt độ hoạt động: 0 °C – 55 °C

#### ***2.1.4 Giới thiệu về cách ly quang Opto 4N35***

4N35 là một opto-coupler sử dụng để cách ly tín hiệu điện giữa hai mạch khác nhau. Thiết bị này có cấu tạo gồm một LED hồng ngoại và một photo-transistor. Khi dòng điện đi qua LED, nó phát ra ánh sáng hồng ngoại. Ánh sáng này kích hoạt photo-transistor, cho phép dòng điện chạy từ Collector đến Emitter của transistor, tạo tín hiệu ở đầu ra. Nhờ cách ly quang học giữa hai phần, 4N35 đảm bảo tín hiệu đầu vào và đầu ra được cách ly hoàn toàn về mặt điện, giúp bảo vệ các mạch khỏi nhiễu và điện áp cao.



Hình 9: Opto 4N35



Hình 10: Sơ đồ mạch opto4N35

1. Anode
2. Cathode
3. No Connection
4. Emitter
5. Collector
6. Base

### **Thông số kỹ thuật 4N35**

Điện áp cách ly: Lên đến 5000 VRMS

Dòng điện LED: Khoảng 10 mA đến 60 mA

Dòng điện đầu ra tối đa: 50 mA (ở đầu ra transistor quang)

Điện áp bão hòa đầu ra (VCE): Thường khoảng 0.3V (ở IC=2 mA)

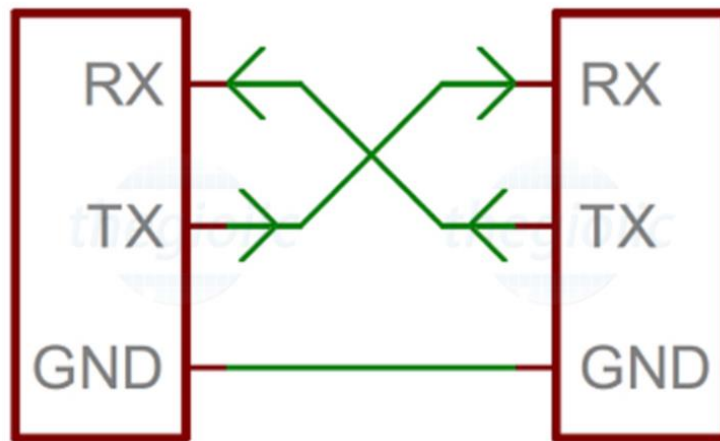
Tần số hoạt động: Tối đa khoảng 300 kHz (ở tốc độ tối đa của transistor)

Nhiệt độ hoạt động: -55°C đến +100°C

### **2.1.5 Các chuẩn giao tiếp được sử dụng**

#### **Chuẩn giao tiếp UART**

UART (Universal Asynchronous Receiver-Transmitter) bộ truyền nhận tín hiệu nối tiếp nội bộ, đây là một trong những giao thức liên lạc giữa thiết bị với thiết bị được dùng phổ biến nhất. Giao tiếp UART được dùng phổ biến trong các thiết bị IoT kết nối với các module như: Wifi, Bluetooth, Xbee, module đầu đọc thẻ RFID với Raspberry Pi, Arduino hoặc vi điều khiển Cisco. Đây cũng là tiêu chuẩn giao tiếp quan trọng và phổ biến trong doanh nghiệp.



Hình 11: Minh họa cách truyền uart

Hai đường dây mà mỗi thiết bị UART sử dụng để truyền dữ liệu đó là:

Transmitter (Tx)

Receiver (Rx)

UART truyền dữ liệu không đồng bộ, có nghĩa là không có tín hiệu đồng hồ để đồng bộ hoá đầu ra của các bit từ UART truyền hoặc để lấy mẫu các bit từ UART nhận. Thay vì tín hiệu đồng hồ, UART truyền thêm các bit start và stop vào gói dữ liệu đã chuyển. Các bit stop xác định điểm bắt đầu và điểm cuối của gói dữ liệu cho phép UART xác định khi nó bắt đầu đọc các bit.

### Cách thức hoạt động của giao tiếp UART

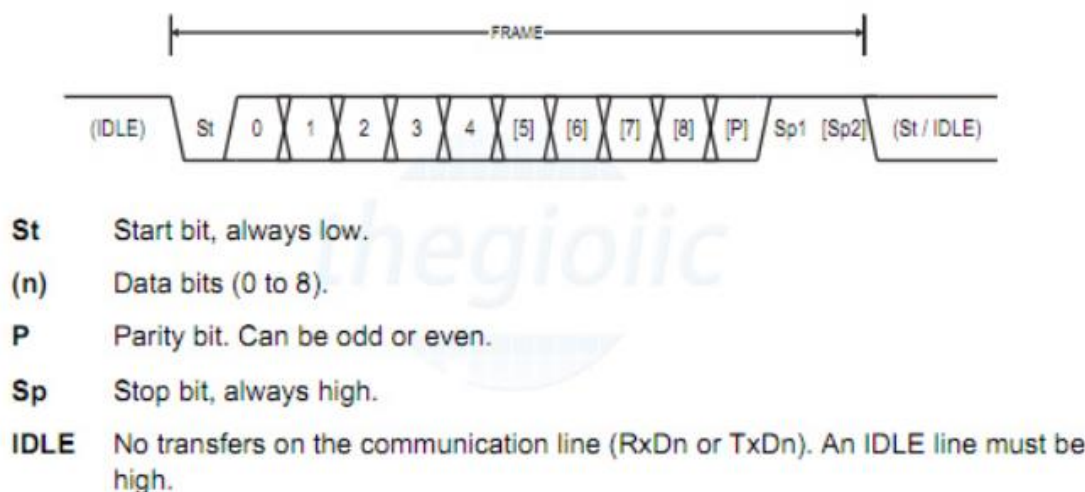
UART truyền dữ liệu thu được từ một bus dữ liệu (Data Bus). Bus dữ liệu được dùng để truyền dữ liệu đến UART thông qua một thiết bị khác như CPU, RAM hoặc vi điều khiển. Dữ liệu được truyền từ bus dữ liệu đến UART truyền dưới dạng nối tiếp. Sau khi UART truyền nhận dữ liệu đồng thời từ bus dữ liệu, nó sẽ thêm một bit start, một bit chẵn lẻ và một bit stop, xuất ra gói dữ liệu. Tiếp theo, gói dữ liệu được tạo ra nối tiếp mỗi bit tại chân Tx. UART nhận chuyển gói dữ



liệu mỗi bit tại chân Rx của nó. UART nhận sau đó chuyển tiếp dữ liệu về trạng thái song song và loại bỏ bit start, bit chẵn lẻ và bit stop. Cuối cùng, UART nhận chuyển tiếp gói dữ liệu song song với bus dữ liệu ở đầu nhận.

UART truyền dữ liệu nối tiếp, theo một trong ba chế độ:

- Full duplex: Giao tiếp đồng thời đến và đi từ mỗi master và slave.
- Half duplex: Dữ liệu đi theo một hướng tại một thời điểm.
- Simplex: Chỉ giao tiếp một chiều.



Hình 12: Cách truyền bit của uart

Dữ liệu UART được tổ chức thành các gói gồm: 1 bit khởi đầu, 5-9 bit dữ liệu, bit chẵn lẻ tùy chọn, và 1-2 bit dừng.

**Bit khởi đầu:** Đường truyền ở mức cao khi không truyền dữ liệu. Để bắt đầu, UART kéo đường truyền xuống thấp trong một chu kỳ.

**Khung dữ liệu:** Gồm 5-9 bit, gửi LSB trước. Có thể dùng 8 bit nếu có bit chẵn lẻ, hoặc 9 bit nếu không dùng bit chẵn lẻ.

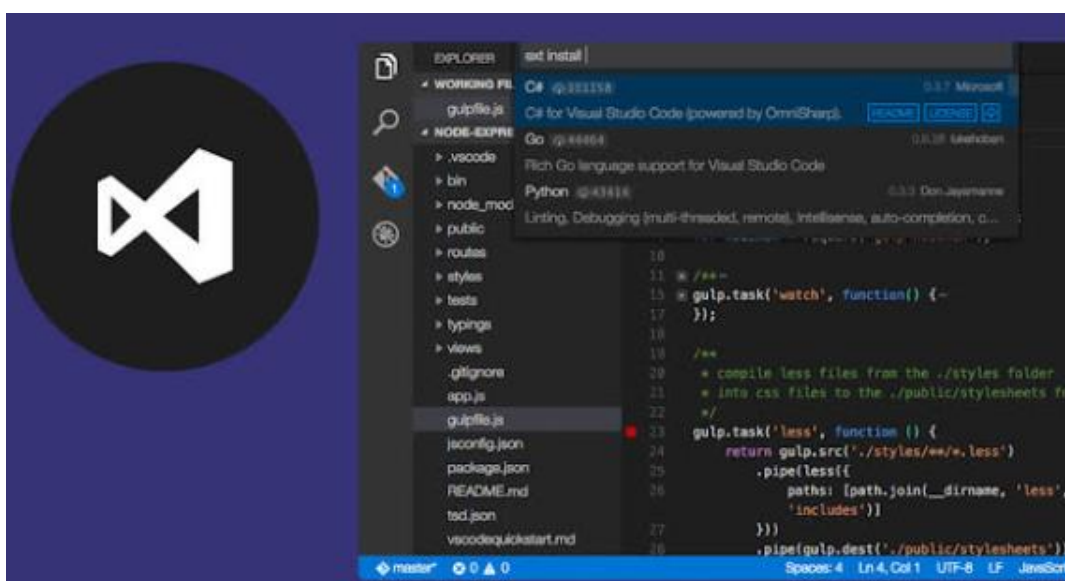
**Bit chẵn lẻ:** Kiểm tra lỗi trong quá trình truyền. Nếu tổng số bit 1 không khớp với cấu hình (chẵn/lẻ), UART biết dữ liệu bị lỗi.

**Bit dừng:** Kết thúc gói dữ liệu bằng cách chuyển đường truyền từ thấp lên cao trong 1-2 bit thời gian.

## 2.2 Phần mềm

### 2.2.1 Giới thiệu về ứng dụng Visual Studio

Visual Studio là một môi trường phát triển phần mềm (IDE) mạnh mẽ do Microsoft phát triển, được ứng dụng phổ biến trong lĩnh vực phát triển ứng dụng, đặc biệt là đối với các ứng dụng Windows. Visual Studio hỗ trợ nhiều ngôn ngữ lập trình, như C #, C + +, Python, JavaScript, VB.NET và nhiều ngôn ngữ khác. Nó cung cấp bộ tài nguyên mạnh mẽ cho phép phát triển, phân tích, kiểm thử và lập trình ứng dụng.

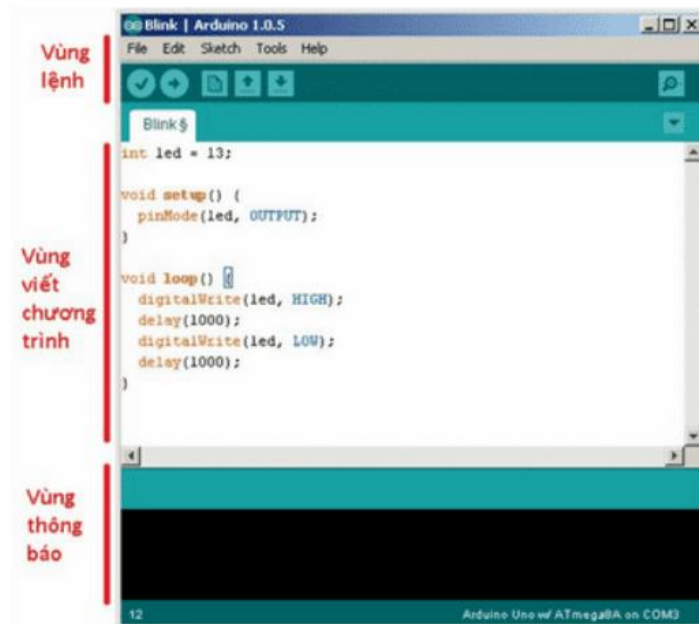


Hình 13: Hình minh họa visul studio

Môi trường phát triển này cũng hỗ trợ việc phát triển các ứng dụng trên nền tảng mobile, web, và desktop, với bộ công cụ hỗ trợ phát triển ứng dụng. NET, Xamarin (cho ứng dụng mobile), và ASP.NET (cho ứng dụng web). Visual Studio có các phiên bản miễn phí và trả phí, với Visual Studio Community là phiên bản miễn phí, thích hợp cho học sinh, sinh viên, và các dự án mã nguồn mở.

### 2.2.2 Giới thiệu về Arduino IDE

Arduino IDE (Integrated Development Environment) là một môi trường phát triển phần mềm mã nguồn mở, được phát triển nhằm giúp người dùng dễ dàng lập trình trên các bo mạch vi điều khiển Arduino. IDE này hỗ trợ tạo mã, biên dịch và tải mã nguồn từ các bo mạch Arduino thông qua cổng USB. Arduino IDE sử dụng ngôn ngữ lập trình C/C++ với một bộ thư viện phong phú, giúp người dùng dễ dàng kết nối với phần cứng và thực hiện các tác vụ phần cứng - phần mềm.



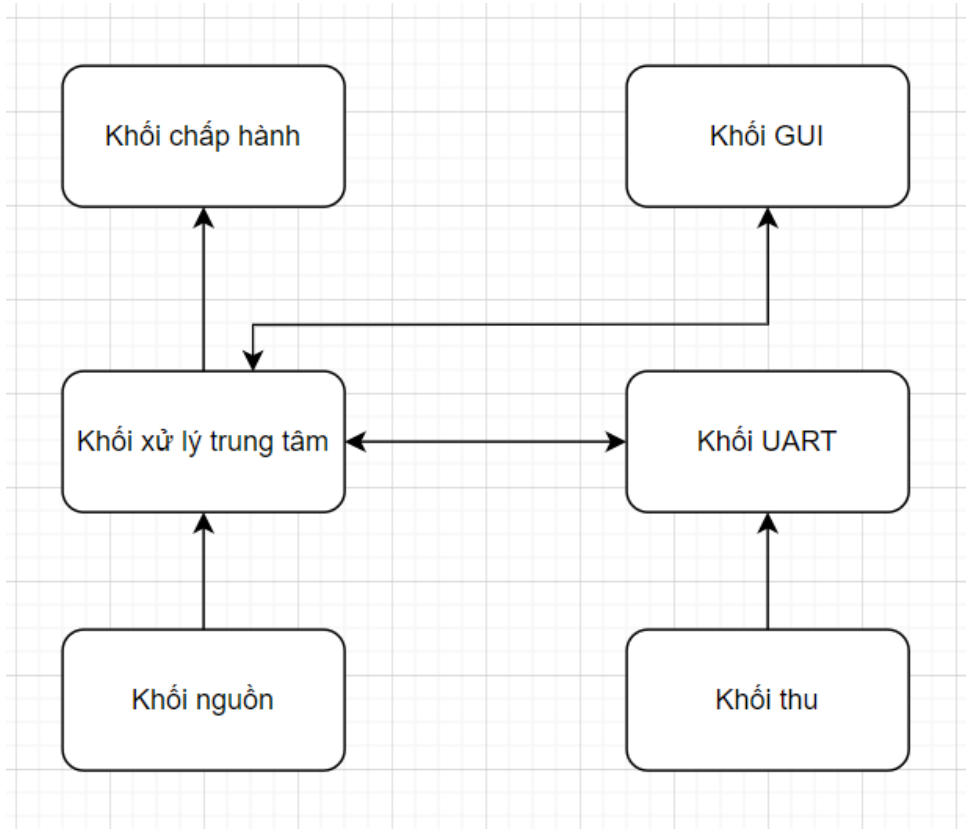
Hình 14: Minh họa giao diện IDE arduino

Arduino IDE cũng hỗ trợ nhiều dòng bo mạch khác nhau, không chỉ giới hạn ở các dòng bo mạch Arduino chính thức mà còn hỗ trợ các bo mạch khác bao gồm ESP32, ESP8266, STM32 với các plugin và thư viện bổ sung. Một tính năng đặc biệt của Arduino IDE là khả năng xây dựng thư viện mã nguồn mở với hàng nghìn thư viện và ví dụ đã được chia sẻ với người sử dụng trên khắp thế giới.

IDE này có thể được cài đặt trên nhiều hệ điều hành, bao gồm Windows, macOS, và Linux, và có phiên bản miễn phí cho người dùng cá nhân cũng như cho các dự án mã nguồn mở.

## CHƯƠNG 3 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 3.1 Thiết kế sơ đồ khối hệ thống



Hình 15: Sơ đồ khối hệ thống

#### **Chức năng các khối:**

Khối nguồn: cung cấp nguồn để các thiết bị có thể hoạt động ổn định.

Khối xử lý trung tâm: chứa Arduino được xem như bộ não của hệ thống. Có chức năng giao tiếp và truyền nhận dữ liệu với khối UART và đưa ra tín hiệu cho khối chấp hành điều khiển Servo

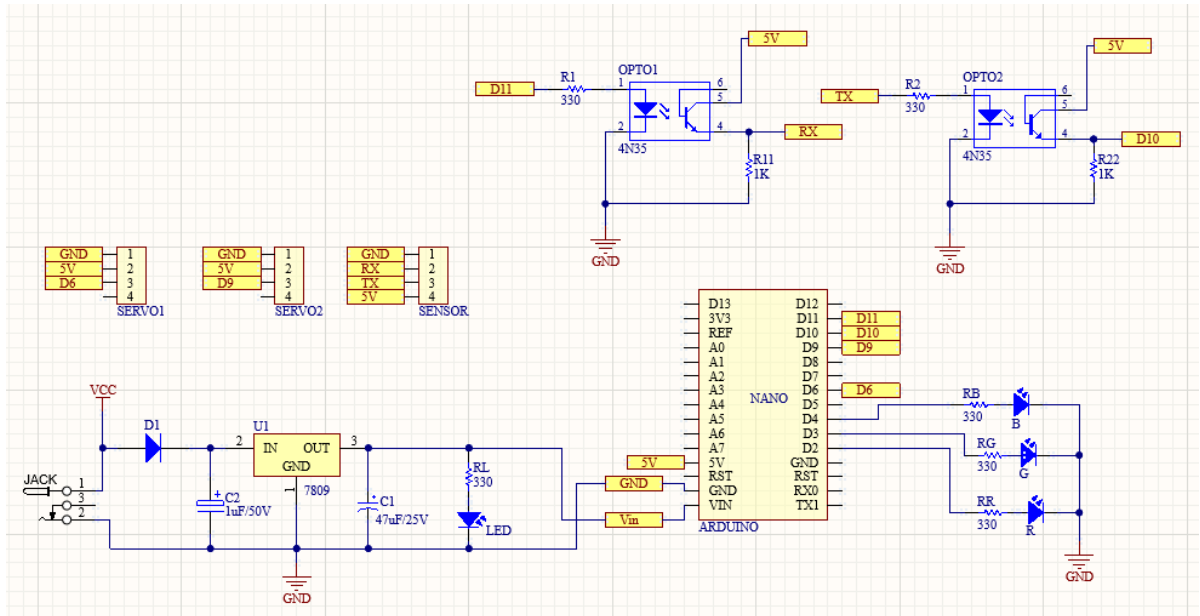
Khối thu: thu thập các tín hiệu từ cảm biến, truyền về khối xử lý trung tâm để xử lý thông qua khối UART.

Khối UART: là khối giao tiếp truyền nhận dữ liệu uart, đóng vai trò làm trung gian.

Khối GUI: đóng vai trò là khối làm giao diện trên ứng dụng visual studio để gửi dữ liệu đưa lên khối chấp hành là Servo để thực thi, thông qua khối xử lý trung tâm là Arduino.

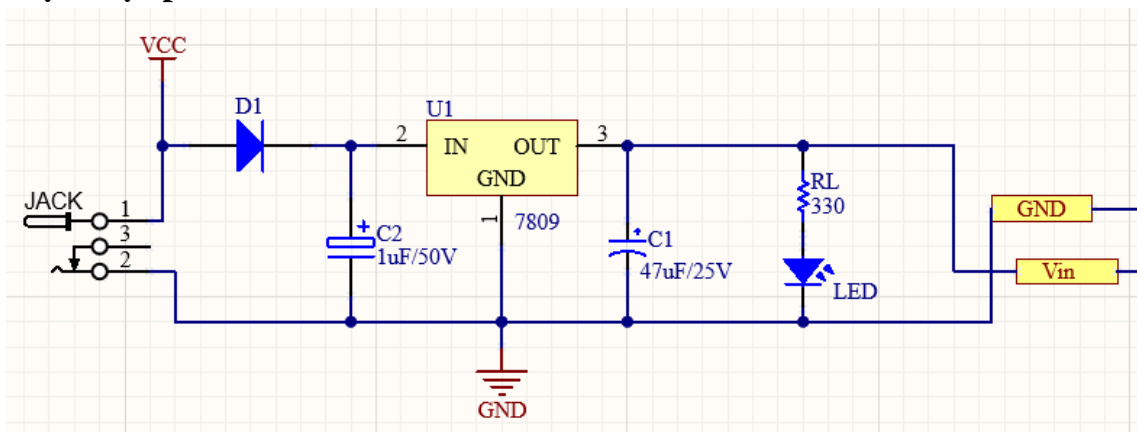
Khối chấp hành: là Servo, nhận dữ liệu từ khối xử lý trung tâm Arduino để thực hiện

### 3.2 Sơ đồ nguyên lý



Hình 16: Sơ đồ nguyên lý của mạch

#### Mạch hạ áp:



Hình 17: Sơ đồ mạch nguồn hạ áp 12V-9V

### 3.2.1 Tính toán thông số mạch

7809 là IC ổn áp tuyến tính, đầu ra ổn định ở 9V. Để đảm bảo hoạt động ổn định, điện áp đầu vào  $V_{in}$  cần thỏa mãn:

$$V_{in} \geq V_{out} + V_{drop} \quad (1)$$

Với  $V_{drop}$  khoảng là 2V

$$V_{in} > 9V + 2V = 11V \quad (2)$$

Từ đó ta cần nguồn  $V_{cc}$  lớn hơn 11V, ta chọn 12V.

Tụ C2 được sử dụng để giảm nhiễu điện áp đầu vào. Ta có giá trị của tụ được chọn theo công thức thực nghiệm:

$$C_{in} = \frac{I_{load}}{10 \cdot f} \quad (3)$$

Với  $I_{load}$ : dòng tải tối đa qua IC

$f$ : tần số nguồn

Ta chọn dòng tải tối đa là  $I_{load} = 2A$ :

$$C_{in} = \frac{2}{10 \cdot 50} = 4000\mu f \quad (4)$$

Tuy nhiên, trong thực tế ta sẽ chọn giá trị nhỏ hơn để xử dụng vì nó lọc nhiễu cao tần tốt hơn.

Tụ C1 được sử dụng ở đầu vào để cân bằng dòng điện và giảm thiểu gợn sóng. Giá trị thường được chọn bởi:

$$C_{out} > 22\mu F \quad (5)$$

Trong mạch hiện tại, ta chọn giá trị phù hợp cho tải thông thường là 47uF.

Điện trở led làm giảm dòng đi vào led, giá trị được tính theo công thức sau:

$$R_L = \frac{V_{out} - V_{LED}}{I_{LED}} \quad (6)$$

Ta có :

$$V_{out} = 9V$$

$V_{LED}$  : điện áp trên led

$I_{LED}$  : Dòng qua led

Ta chọn  $I_{LED} = 15mA$

$$R_L = \frac{9V - 2V}{0.015V} = 466.67\Omega \quad (7)$$

Ngoài thực tế , với ý đồ mong muốn LED sáng hơn để có thể theo dõi tốt hơn, nên ta chọn 330 ôm.

Diot bảo vệ mạch khỏi trường hợp cấp nguồn sai cực:

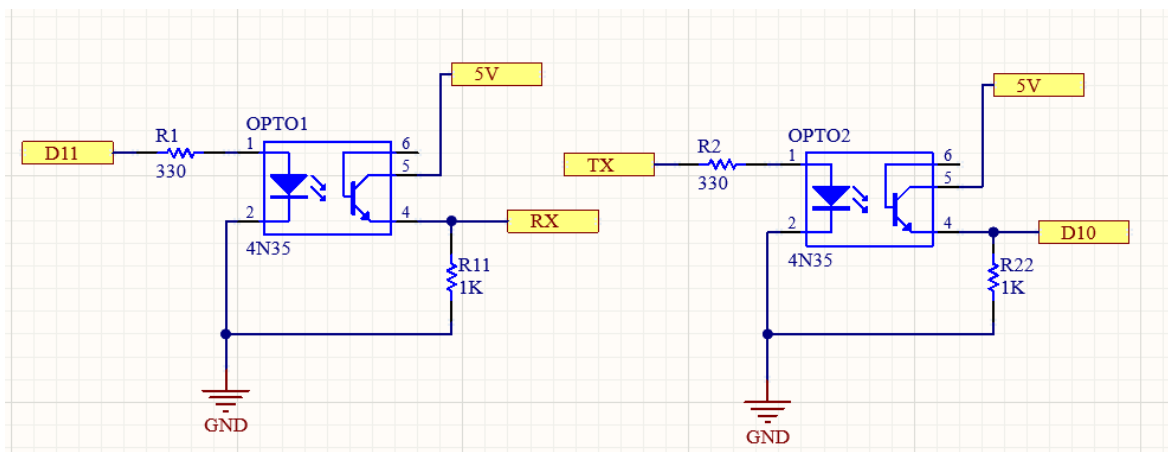
- Điện áp ngược chiều  $V_R \geq V_{in}$
- Dòng tối đa  $I_f \geq I_{load}$

ta chọn dòng tải tối đa là 2A và nguồn vào là 12V, ta chọn Diot 1N4007

$$V_R = 1000V, I_f = 2A$$

**Mạch opto 4N35:**





Hình 18: Sơ đồ nguyên lý opto 4N35

### 3.2.2 Tính toán thông số mạch

R1 và R2 giới hạn dòng qua LED trong opto-coupler (4 N 35). Dòng qua LED được xác định theo công thức:

$$R = \frac{V_{CC} - V_{LED}}{I_{LED}} \quad (7)$$

$$V_{CC} = 5V$$

$V_{LED}$  : điện áp trên led bên trong 4N35

$I_{LED}$  : Dòng qua led

Ta chọn dòng 10mA

$$R_1 = \frac{5V - 1.2V}{0.01A} = 380\Omega \quad (8)$$

Trong mạch, giá trị gần chính xác là 330Ω để có dòng điện cao hơn một chút, đảm bảo kích hoạt phototransistor.

R11 và R22 được dùng làm điện trở nâng lên (pull-up) ở đầu ra của opto-coupler. Giá trị được lựa chọn phụ thuộc vào trở kháng đầu vào của thiết bị tiếp nhận và dòng điện cần thiết để nâng đầu ra lên mức cao.

Ta có đầu ra là  $V_{out} = 5V$  và dòng tiêu thụ thấp

$$R = \frac{V}{I} = \frac{5V}{0.001A} = 5k\Omega \quad (9)$$

Trong mạch, giá trị  $1k\Omega$  được dùng nhằm đảm bảo thời gian đáp ứng nhanh hơn khi phototransistor chuyển trạng thái.

4N35 là opto-coupler có LED và phototransistor bên trong. Thông số cần chú ý:

$V_F$  điện áp rơi trên led

$I_{LED}$ : dòng qua LED

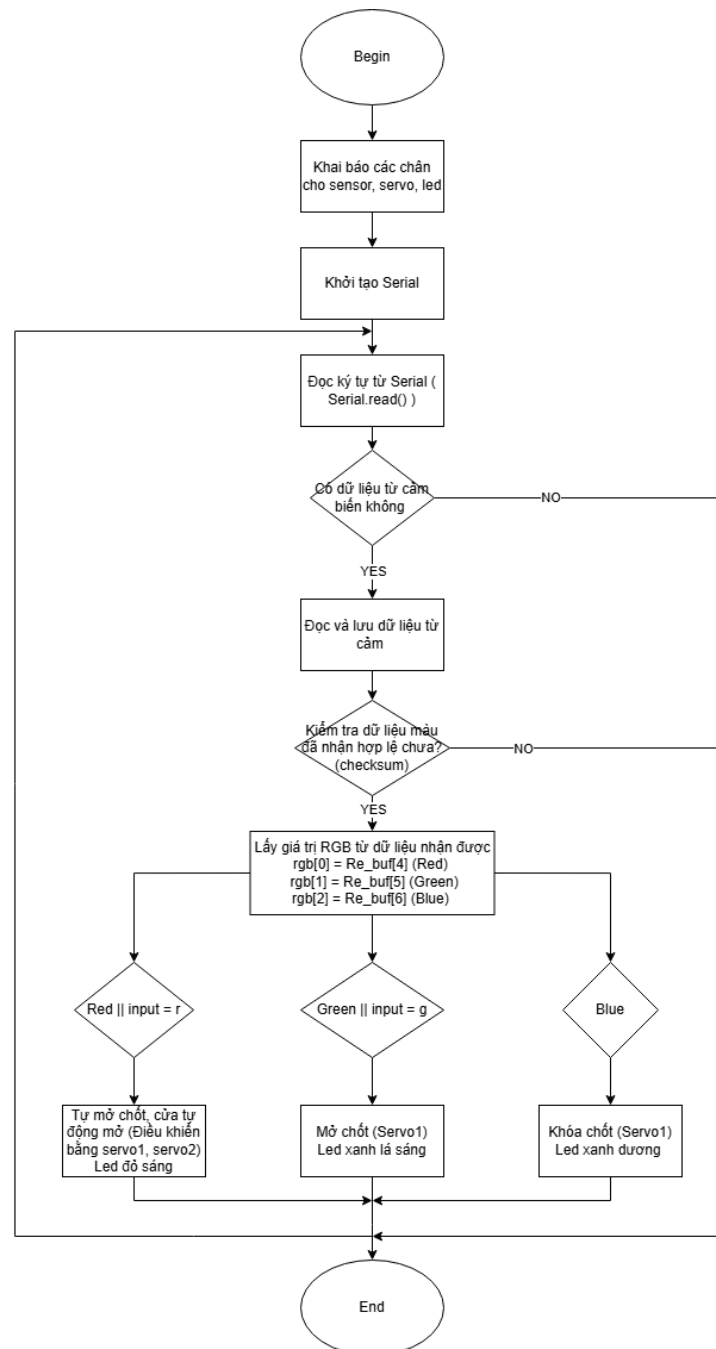
CTR tỷ lệ truyền dòng, thường từ 50% đến 200%

Tính toán dòng qua phototransistor

Ta chọn I là 10mA, CTR là 100%

$$I_C = I_{LED} > CTR = 10mA \cdot 1 = 10mA \quad (10)$$

### 3.3 Lưu đồ giải thuật

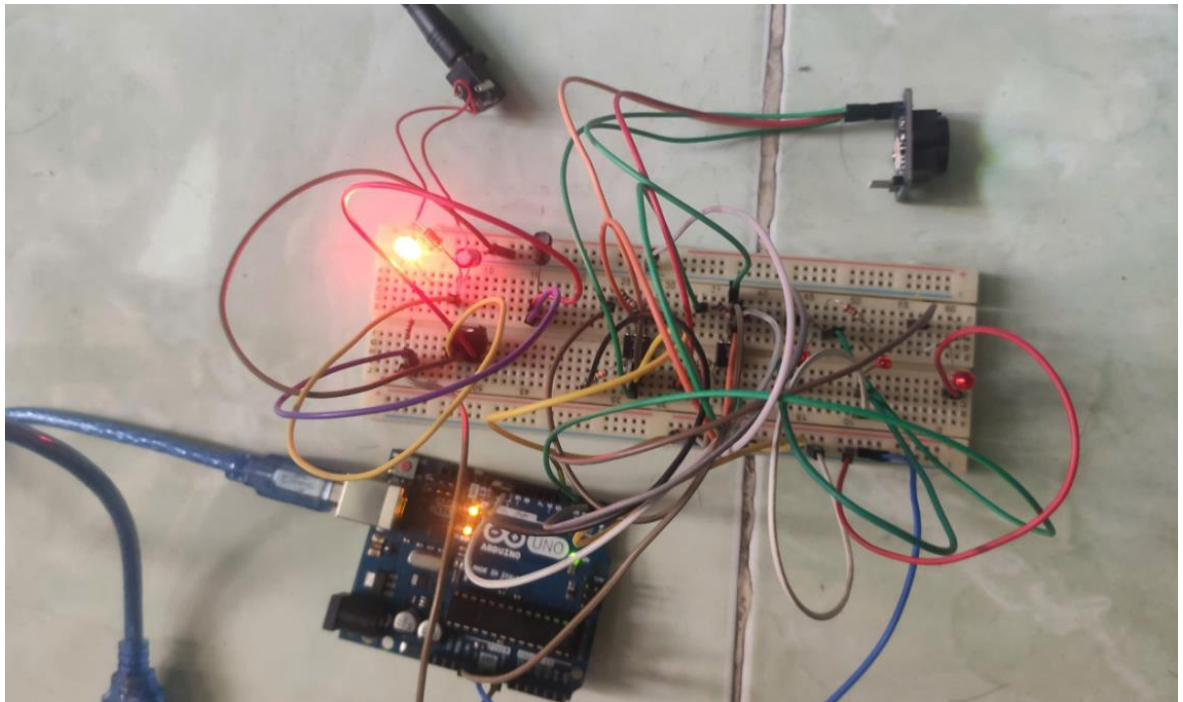


Lưu đồ giải thuật là một trong những điều rất quan trọng để có thể đảm bảo mọi người có thể hiểu được cách hoạt động logic của mạch, ở trên là lưu đồ hoạt động cho toàn bộ dự án.

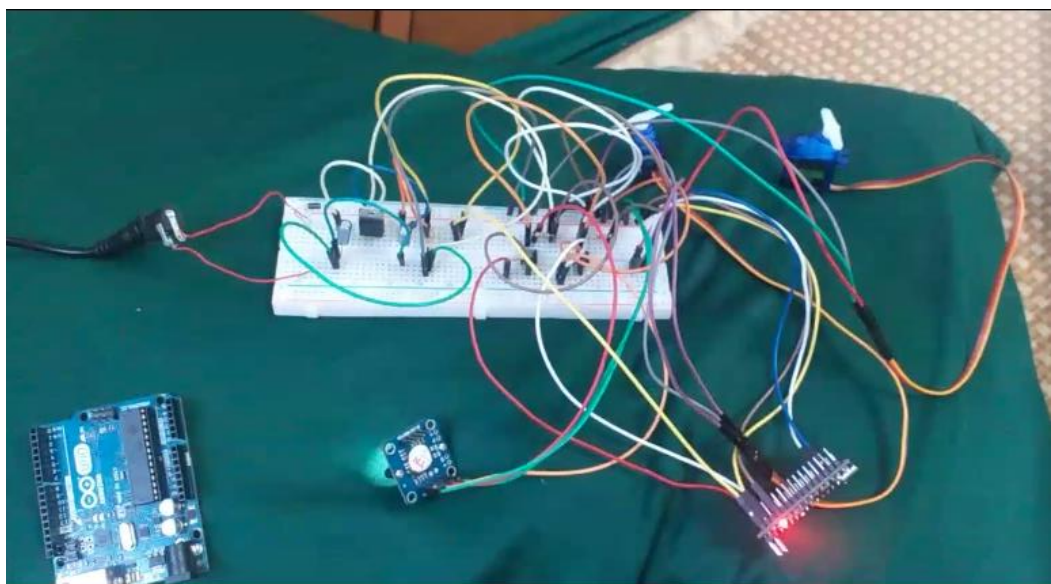
### 3.4 Test bo mạch

Một phần không kém quan trọng đó chính là test mạch thiết kế trước khi thiết kế ra sản phẩm hoàn thiện, việc này sẽ đảm bảo cho mạch không xảy ra sai sót cũng như những lỗi kỹ thuật không mong muốn xảy ra, hạn chế được những rủi ro không mong muốn.

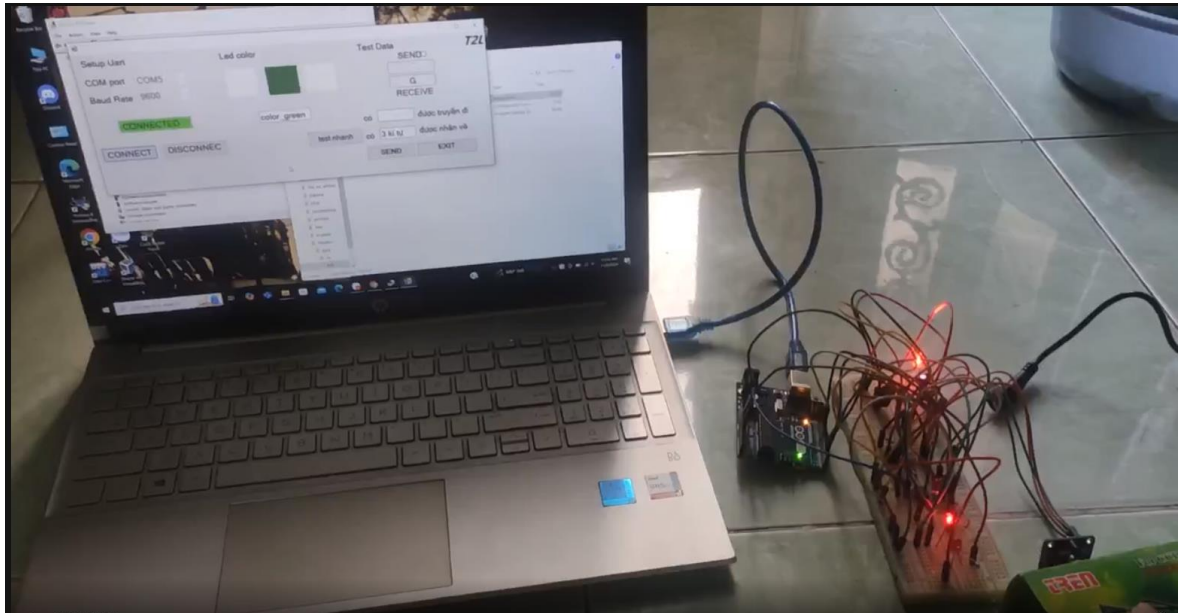
#### 3.4.1 Ảnh test bo mạch thực tế



Hình 20: Test truyền uart giữa cảm biến và arduino



Hình 19: test board ,mạch với các thiết bị ngoại vi



Hình 21: Test board mạch và giao diện C#

### 3.5 Thiết kế mạch PCB trên ứng dụng Altium

Một trong những yếu tố quan trọng nhất đó chính là thiết kế mạch PCB để có thể gia công mạch cứng, là phần để liên kết các linh kiện lại với nhau thành 1 thể thống nhất và logic để mạch hoạt động đúng yêu cầu đã đề ra.

#### 3.5.1 Tính toán mạch PCB

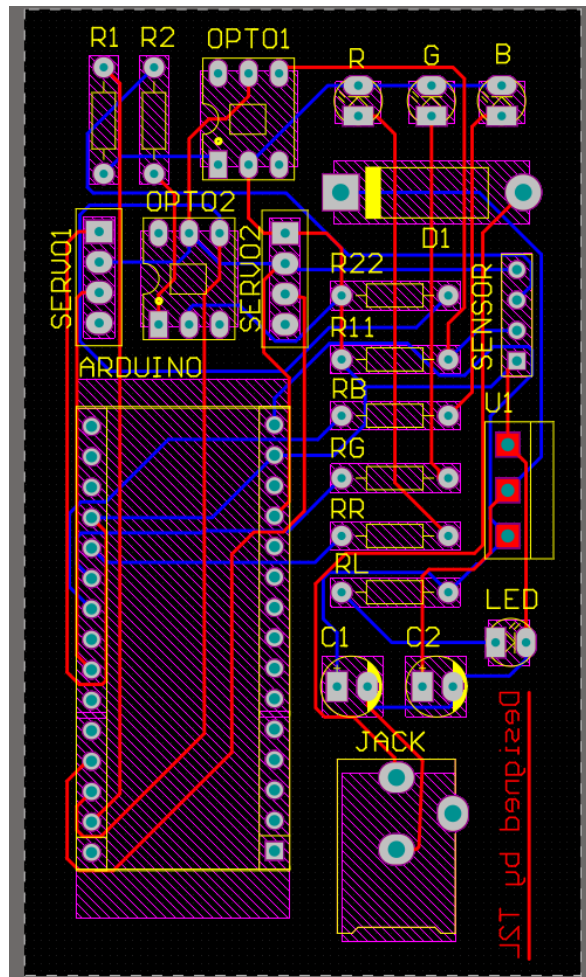
Với những yêu cầu ban đầu đã đề ra, cũng như bối cảnh dự án ta cần thiết kế mạch PCB với những tiêu chuẩn sau:

- Thiết kế bố trí linh kiện gọn gàng để tối ưu mạch nhất có thể
- Tập trung những linh kiện cùng loại thành 1 chỗ,
- Thiết kế led góc dễ nhìn để quan sát trạng thái trực tiếp
- Mạch được khắc logo của team để dễ nhận diện

- Mạch có kích thước là 5 x 8 cm, đảm bảo nhỏ gọn nhưng vẫn hoạt động trơn tru
- Dây đi trên 2 lớp có độ rộng là 0.2mm
- Lỗ via là 0.7mm

Những tiêu chuẩn đã được thảo luận và xem xét bao quát bối cảnh dự án. Ngoài ra, để thuận tiện cho các thiết bị ngoại vi ta sẽ thiết kế các đầu ra của các kết nối ngoại vi là các jack, bus hoặc các loại cắm để có thể nối dài ra ngoài nhằm phục vụ cho việc thiết kế mô hình.

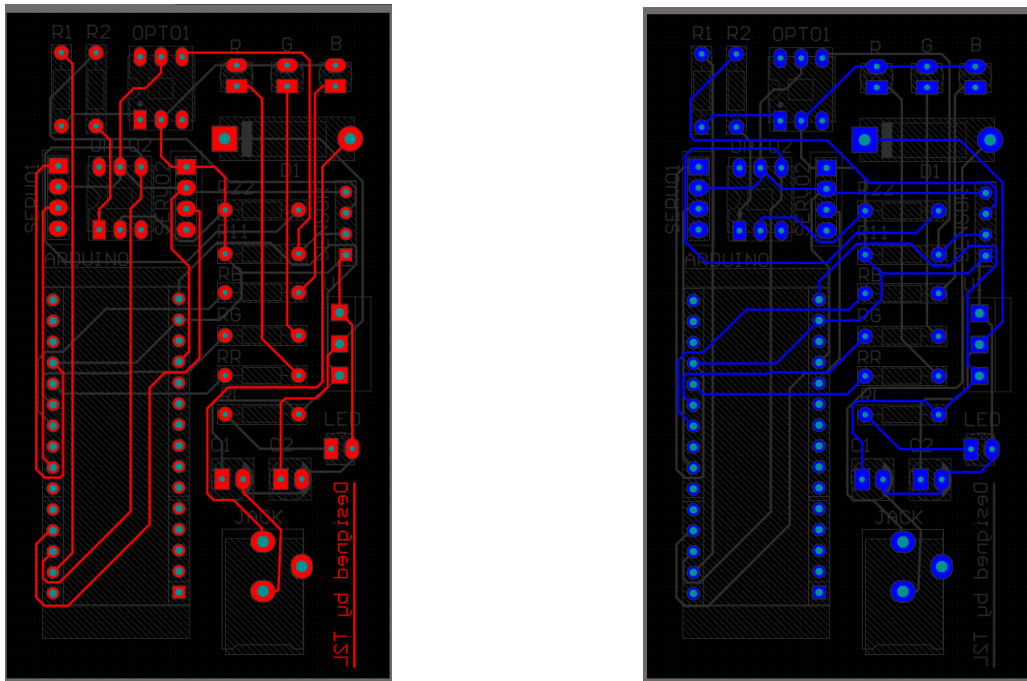
### 3.5.2 Mô hình 2D trên hai lớp:



Hình 22: Bố trí linh kiện và đi dây 2 lớp

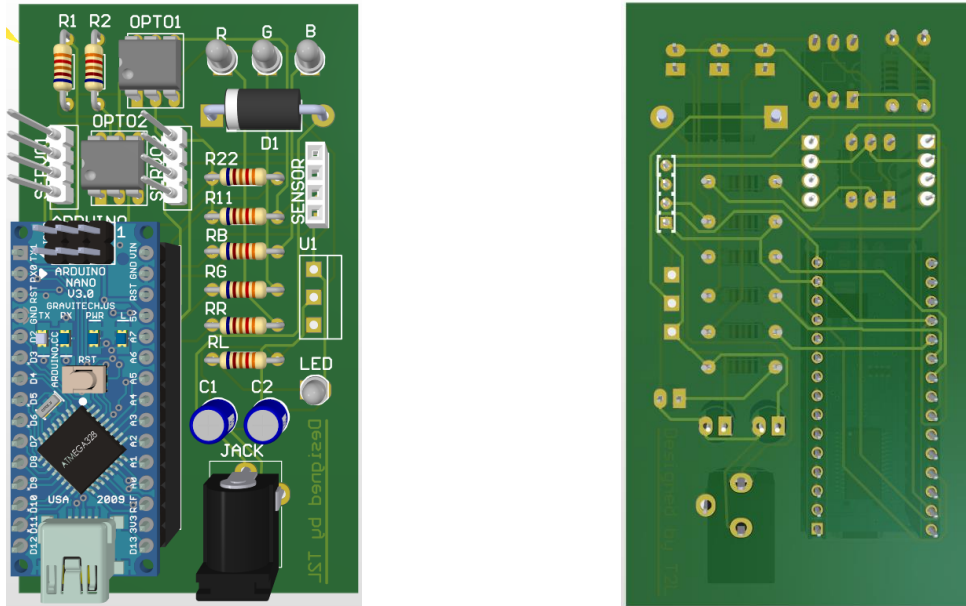


### 3.5.3 Mô hình 2D của lớp Top và lớp Bottom:



Hình 23: Ảnh đi dây PCB của lớp top và bottom

### 3.5.4 Mô hình 3D của mạch:



Hình 24: Ảnh 3D của mạch PCB

### 3.6 tính toán tốc độ truyền uart

trong sản phẩm, ta chọn tốc độ truyền là 7200 baud,ta sẽ tính toán tốc độ truyền của uart như sau:

$$T_{bit} = \frac{bit}{Baudrate} \quad (11)$$

thời gian 1 bit được tính bằng công thức sau:

$$T_{bit} = \frac{1}{7200} giây = 0.00013889 giây = 138.89\mu s \quad (12)$$

Vậy 1 bit sẽ mất khoảng 138.89  $\mu s$

Ta có thời gian của 1 bit thì ta sẽ suy ra được thời gian của 8 bit là :

$$T_{8bit} = T_{bit} \times 8 = 138.89 \times 8 = 1.11112ms \quad (13)$$

Vậy thời gian để truyền 8 bit đi là 1.11112ms

Nếu ta tính thêm cả bit start/stop(truyền không đồng bộ) , ta sẽ có thời gian như sau:

$$T_{10bit} = T_{bit} \times 10 = 138.89 \times 10 = 1.3889ms \quad (14)$$

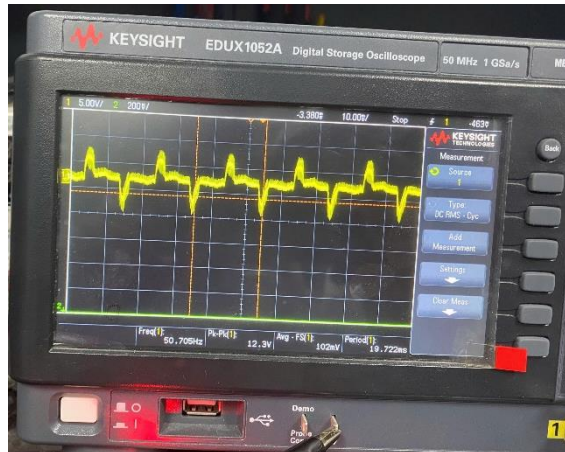
Việc tính toán các thông số là cực kì quan trọng trong bước thiết kế sản phẩm,điều này là cần thiết để đảm bảo mạch hoạt động đúng với kì vòng cũng như các tiêu chí kĩ thuật một cách hoàn hảo.



## CHƯƠNG 4 THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

### 4.1 Hình ảnh thực tế đo bằng oscilloscope

#### 4.1.1 Sóng nguồn đi vào



Hình 25: nguồn vào oscilloscope

Đây là dạng sóng đo đầu vào từ adapter vào board mạch, từ trên hình ta thấy được các thông số sắp sỉ bằng các thông số được ghi từ adapter như tần số cũng như nguồn ra.

So sánh thông số đo đạc và thông số của nhà sản xuất:

Thông số	đo bằng oscilloscope	Thông số nhà sản xuất
Freg(tần số)	50.705Hz	47-63Hz
Pk-pk	12.3v	12v

Và trên hình ảnh còn một vài thông số cơ bản khác như:

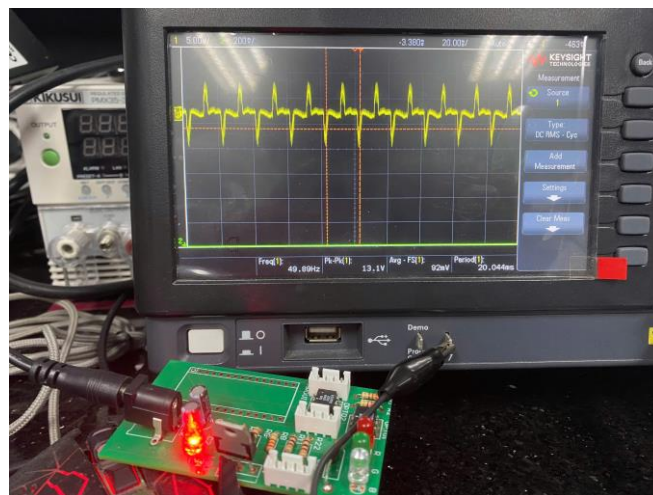
- Avg-FS(giá trị trung bình của tần số lấy mẫu): 102mV
- Period(chu kì tín hiệu):19.722ms

Kết luận: Ta thấy được thông số đo và thông tin nhà sản xuất là giống nhau.

#### 4.1.1.1 Một vài tấm ảnh thêm về nguồn vào

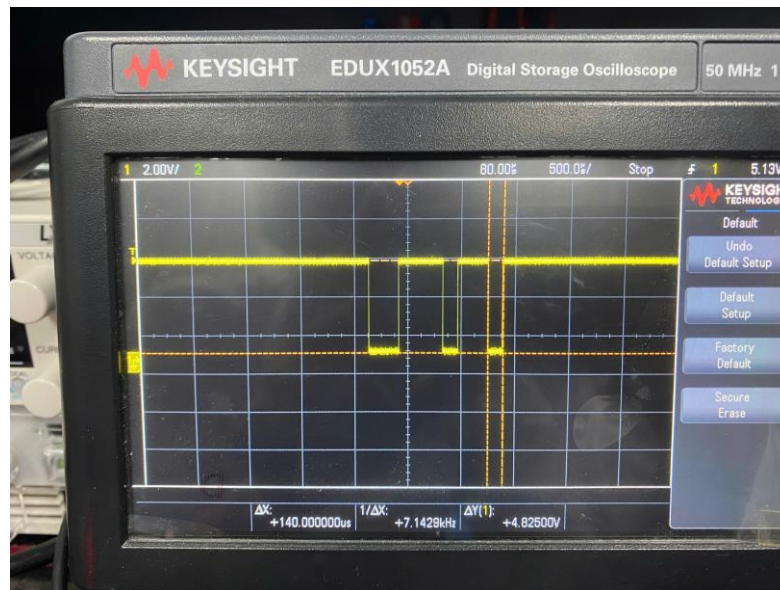


Hình 26: 1 chu kỳ của nguồn vào



Hình 27: ảnh đo thực tế

#### 4.1.2 Đo tín hiệu uart của mạch bằng oscilloscope



Hình 28: sóng đo truyền tín hiệu uart



Hình 29: truyền kí tự từ giao diện sang mạch bằng uart

Từ những tính toán về tốc độ truyền ở trên, ta có tốc độ truyền đi của 1 bit với baud 7200 là  $138.89\mu s$ . ta dùng oscilloscope đo được và kiểm tra độ rộng của bit đầu tiên thấy rằng.

Thông số đo của ảnh trên:

$$\Delta x = 140\mu s$$

$$1/\Delta x = +7.1429KHz$$

$$\Delta Y = +4.82500V$$

$\Delta x = 140\mu s$  nghĩa là độ rộng từ x1 đến x2. Ta thấy được sự chênh lệch của tính toán và đo thực tế của 1 bit là :

$$\Delta x - T_{bit} = 140\mu s - 138.89\mu s = 1.11\mu s \quad (15)$$

Vậy sự chênh lệch giữa lý thuyết và thực tế gần như là quá nhỏ, cho nên ta kết luận rằng tín hiệu truyền đi đúng chính xác gần như tuyệt đối.

$1/\Delta x = +7.1429KHz$  là tần số tín hiệu hiển thị, chỉ số này phù hợp với công thức  $T = \frac{1}{f}$ .

$\Delta Y = +4.82500V$  là điện áp xung tính từ Y1 đến Y2, thông số này hoàn toàn chính xác so với điện áp đầu ra của uart trên arduino nano là +5v.

Ta thấy được tín hiệu của uart truyền đi kí tự “n” có binary là 01001110, có 8 bit data và 2 bit start/stop với bit start bằng 0 và stop bằng 1. Bit đầu tiên từ trái sang là 0 sau đó là chuỗi data và kết thúc bằng 1. Lưu ý, chuỗi data phải đọc từ bên phải sang là 01110010 vì 0111 là trong số cao và 0010 là trong số thấp, từ đó ta thấy được trên hình ảnh đã truyền đúng bit. Ta biết 1 bit sắp xỉ là  $140\mu s$  nên để biết những sóng vuông chưa bao nhiêu bit ta có thể đo bằng chức năng X1 và X2, rồi lấy thông số  $\Delta x$  mới so sánh với  $\Delta x$  cũ bằng cách cộng  $\Delta x$  cũ lên sẽ suy ra được số bit trong 1 xung. Vậy kí tự được truyền đi đúng với mong đợi.

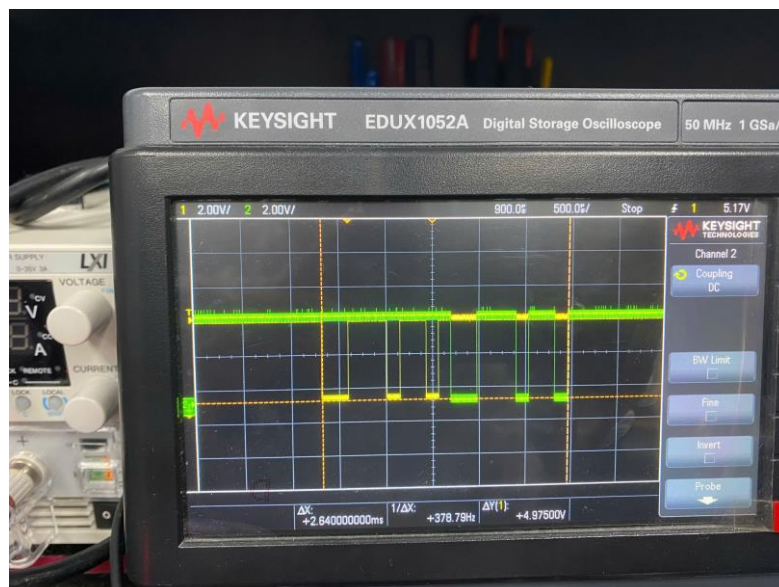


Hình 30: độ rộng của tín hiệu TX với 10 bit



Hình 31: độ rộng của tín hiệu RX với 10 bit





Hình 32: độ dài của nhận về và truyền đi của uart(TX/RX)

ta có các thông số của hình 30 và 31 như sau:

thông số	hình 30	hình 31
$\Delta x$	+1.26ms	1.25ms
$1/\Delta x$	793.65Hz	800Hz
$\Delta Y$	4.825V	5V

Như vậy, nếu so sánh từ bảng trên thì ta thấy được độ dài của tín hiệu khoảng 1.25ms , từ lí thuyết ta có thể tính ra sai số là:

$$\Delta x \text{ sai số} = 1.25ms - 1.3889ms = -0.1389ms \quad (16)$$

Từ thông số trên ta thấy được sai số là rất nhỏ và kết luận rằng tín hiệu truyền đi hoàn toàn chính xác so với lý thuyết tính toán cũng như truyền đi và nhận về là hoàn toàn trùng khớp.

2 thông số còn lại cùng bằng nhau và đúng với những tính toán lý thuyết và thông số thực tế.

Thông số hình 32 như sau:

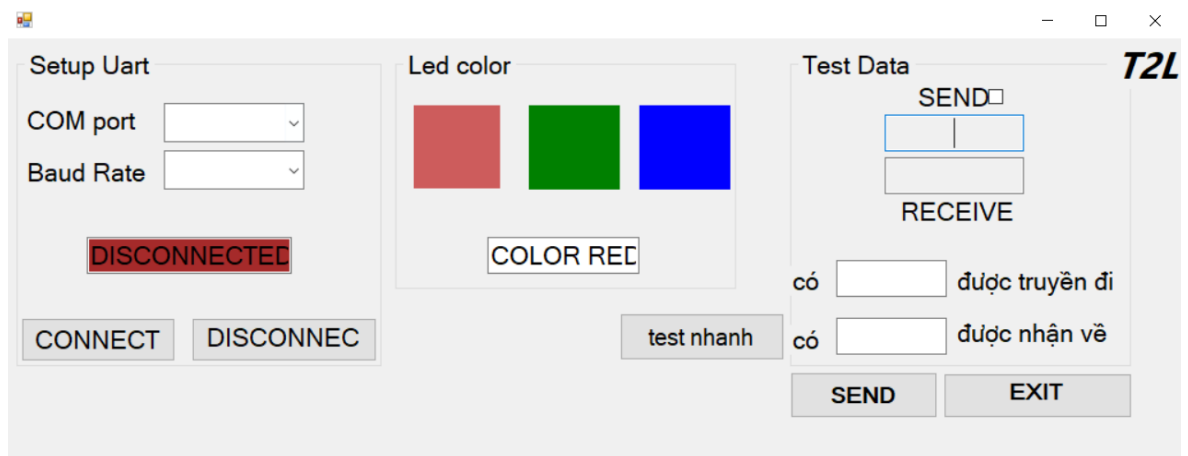
thông số	hình 31
$\Delta x$	+2.64ms
$1/\Delta x$	378.79Hz
$\Delta Y$	4.975V

Ta có độ rộng của tín hiệu là 1.25ms, vậy suy ra 2 tín hiệu là 2.5ms, ở hình 31 ta thấy được độ rộng tín hiệu là 2.64ms, vậy sai lệch là khoảng 0.14ms.

## 4.2 Thiết kế giao diện C#

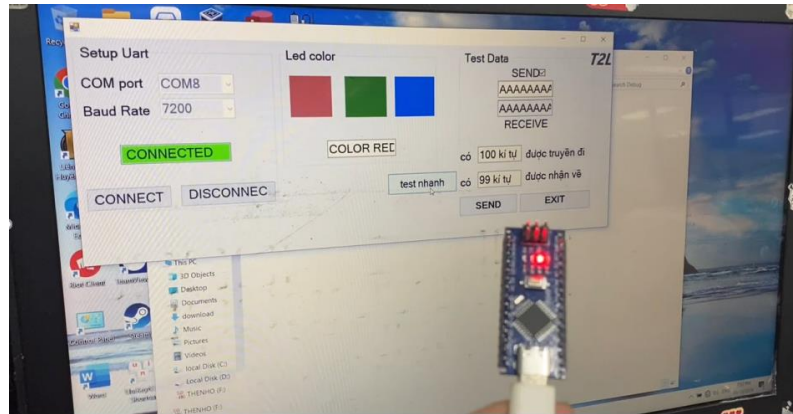
Một trong những mục quan trọng không thể thiếu đó chính là giao diện để hiển thị trạng thái cũng như truyền tín hiệu uart, giao diện sẽ bao gồm các chức năng như sau:

- Chọn cổng com và lựa chọn baud
- Kết nối và hủy kết nối
- Hiển thị màu sắc từ cảm biến
- Test data truyền nhận qua uart
- Chức năng text nhanh 100 kí tự A và hiển thị số kí tự truyền đi và nhận về



Hình 33: giao diện C#

#### 4.2.1 Một vài minh chứng về truyền nhận uart



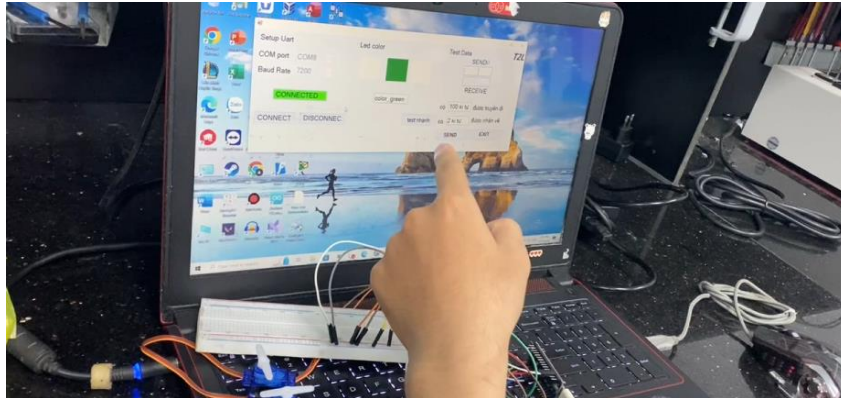
Hình 34: truyền 100 kí tự A

Tỉ lệ truyền và nhận qua uart gần như là 100% với 100 kí tự A.

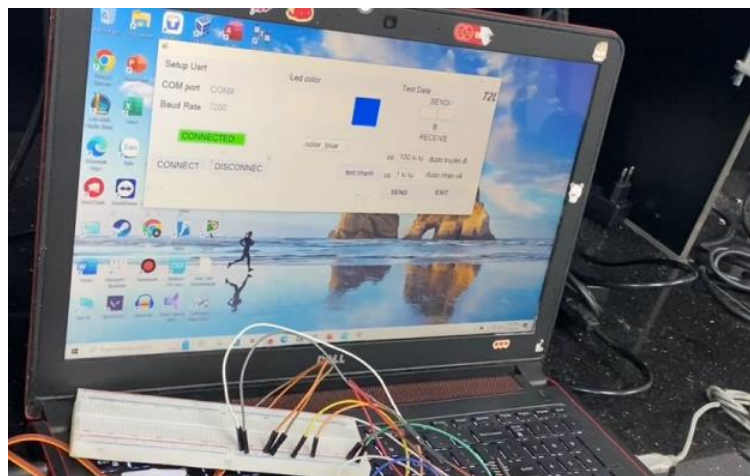


Hình 35: hiển thị màu đỏ từ cảm biến





Hình 36:hiển thị màu xanh lá từ cảm biến



Hình 37:hiển thị màu xanh dương từ cảm biến

ở trên là những minh chứng được thực hiện thực tế kiểm tra giao tiếp uart giữa cảm biến và arduino nano,tốc độ truyền nhanh và ổn định.

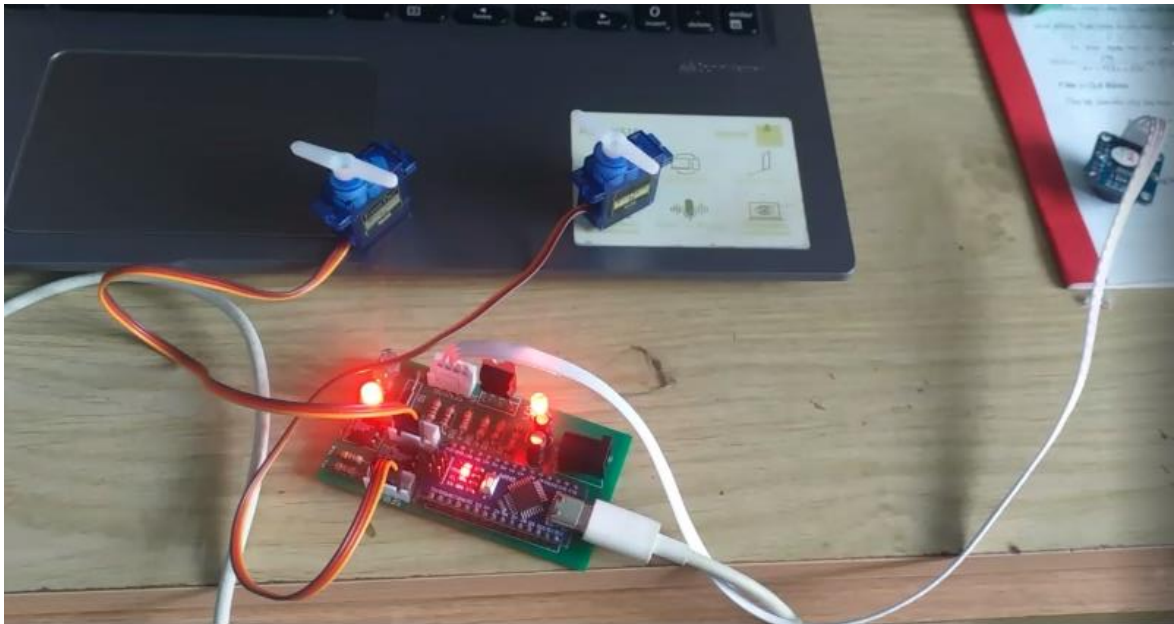
### 4.3 Ảnh thực tế của mạch

Sau quá trình thiết kế thì mạch sẽ được gia công , có 2 cách gia công mạch đó là thực hiện gia công tay ở nhà bằng các phản ứng hóa và lý, nhưng để nâng cao độ thẩm mỹ và tính chuyên nghiệp cho mạch , chúng ta sẽ thực hiện đặt mạch tại bên thứ 3, và đây là kết quả.

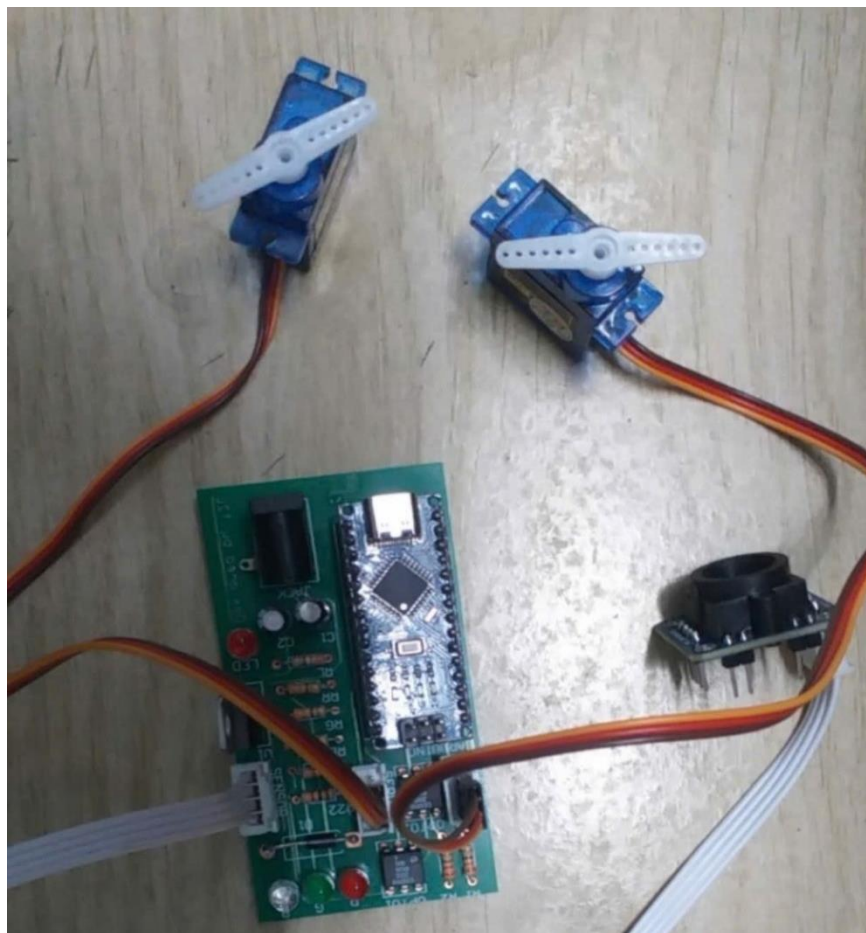


Hình 38: Ảnh mạch PCB thực tế

#### 4.3.1 Hình ảnh kết nối mạch với các thiết bị ngoại vi



Hình 40: cấp nguồn và chạy thử



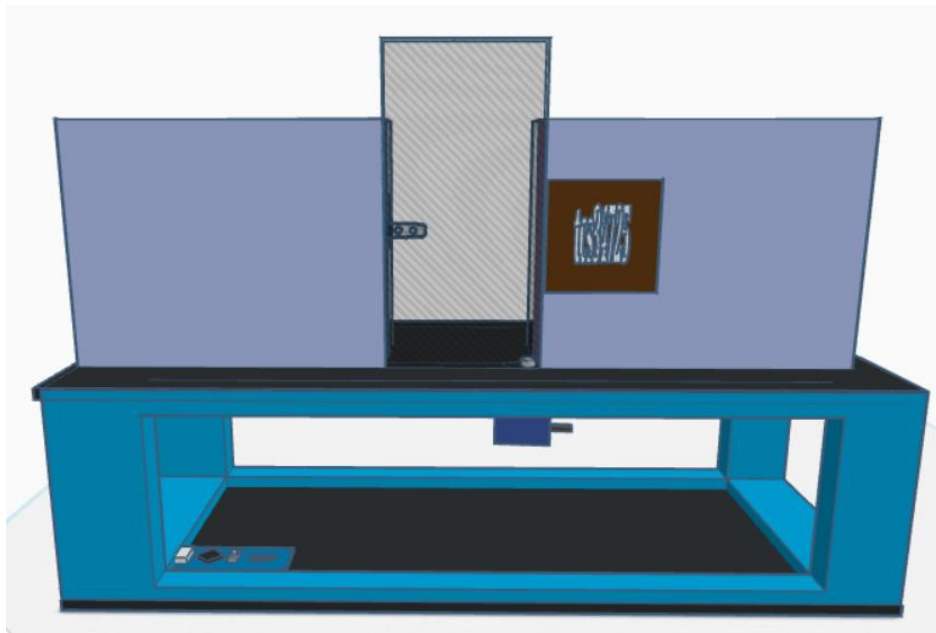
Hình 41: kết nối với các thiết bị ngoại vi

Hình 42: Ảnh chạy mạch

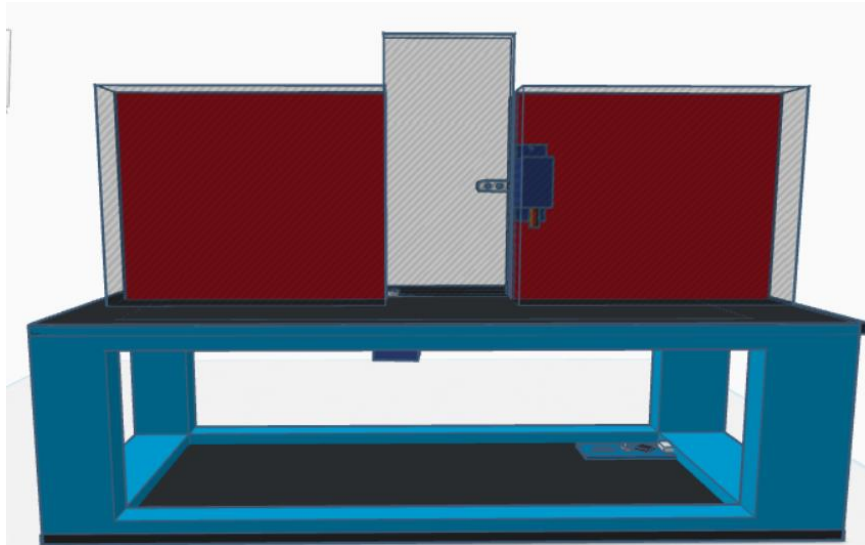
#### 4.4 Mô hình cửa

Để có thể trừu tượng được thực tế của ứng dụng sản phẩm cũng như có cái nhìn rõ ràng nhất, chúng ta sẽ thiết kế mô hình sản phẩm nhà với cửa và cách bố trí mạch gần sát nhất với thực tế.

##### 4.4.1 Ảnh mô hình mô phỏng bằng tinkercad



Hình 43: ảnh mặt trước của mô hình



Hình 44: ảnh sau của mô hình

Để có 1 góc nhìn tốt cũng như định hướng thiết kế mô hình, ta sẽ sử dụng tinkercad để vẽ mô hình 3D để có thể quan sát và thực hiện sản phẩm.

#### ***4.4.2 Ảnh thực tế sản phẩm mô hình***

Sản phẩm mô hình được thiết kế dựa trên mô phỏng tinkercad và được thiết kế lại đôi chút để phù hợp với thực tế, sản phẩm được hoàn thiện bằng format.





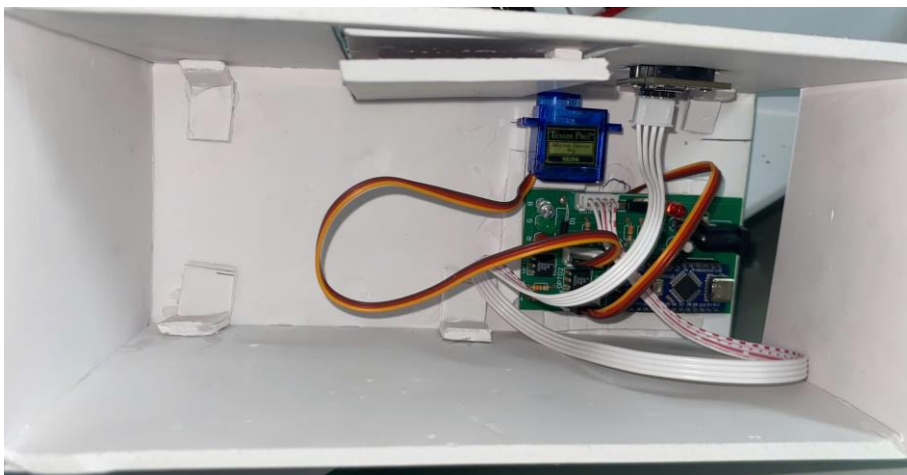
Hình 45: Quá trình thi công sản phẩm



Hình 46: Mặt trước mô hình



Hình 47: Mặt bên cạnh mô hình



Hình 48: Bên trong mô hình

Mô hình hoạt động đúng với kết quả mong đợi, các thiết bị ngoại vi hoạt động đúng với chức năng ban đầu đã đề ra.

#### **4.5 Đánh giá kết quả đạt được**

Board mạch được hoàn thiện và gần như không có lỗi vặt gì trong quá trình hoạt động, cùng với giao diện được hoàn thiện bằng C#, sản phẩm đã hoạt động ổn định với những mục tiêu đã được đề ra

Trong quá trình thực hiện, những kinh nghiệm quý báu đã được rút ra sau sản phẩm trên là hãy nhớ để ý đến dây đi trên mạch in, mạch tích khá là nhỏ nên hãy thiết kế to hơn chút tùy từng dự án mà bạn muốn làm ra, hãy luôn test mạch trước khi hoàn thiện sản phẩm PCB vì đó là điều cực kì cần thiết để tránh những lỗi đáng tiếc sau này.

### **KẾT LUẬN VÀ KIẾN NGHỊ**

Là sản phẩm thiết kế cho cánh cửa tự động áp dụng cảm biến màu sắc, gần như trên thị trường chưa được áp dụng nhiều, sản phẩm đã có thể nhận diện màu sắc và điều khiển cánh cửa theo chương trình đề ra, nhưng vẫn còn một số giới hạn như



khoảng cách mà cảm biến có thể đo được, hay là giao diện còn cần phải dùng dây để cập nhật trạng thái. Mong rằng trong tương lai có thể cải tiến được việc dữ liệu hiện hóa trên giao diện có thể được thành không dây và trên mobile để tiện lợi hơn trong cuộc sống hằng ngày.

## TÀI LIỆU THAM KHẢO

- [1] Docx.com.vn, "Hệ thống phân loại cà chua theo màu sắc," . [Online].  
Available: <https://docx.com.vn/tai-lieu/do-an-tot-nghiep-he-thong-phan-loai-san-pham-theo-mau-sac-460><https://docx.com.vn/tai-lieu/do-an-tot-nghiep-he-thong-phan-loai-san-pham-theo-mau-sac-460>.
- [2] doantotnghiep.vn, "hệ thống phân loại sản phẩm theo màu sắc," [Online].  
Available: <https://doantotnghiep.vn/do-an-thiet-ke-he-thong-dieu-khien-tu-dong-he-thong-phan-loai-san-pham-theo-mau-sacv>.