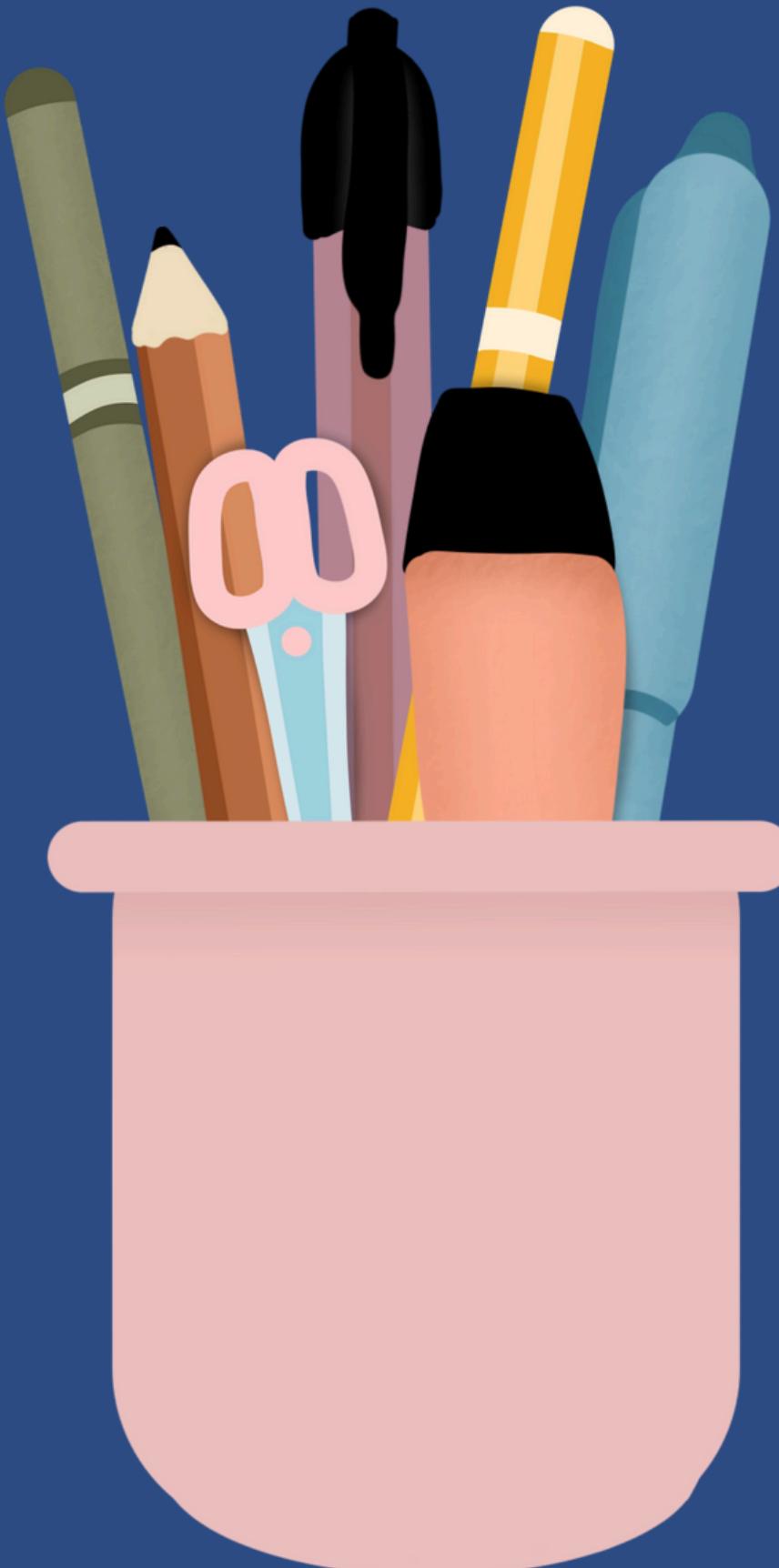




ĐỒ ÁN MÔN HỌC THIẾT KẾ HỆ THỐNG NHÚNG

HỆ THỐNG KIỂM SÓÁT XE TRONG BÃI ĐÔ XE

NGUYỄN MINH LỘC - 21522293
NGUYỄN THÀNH TÀI - 21522568



I. GIỚI THIỆU

- 1/ Đặt vấn đề
- 2/ Mục tiêu

II. CƠ SỞ LÝ THUYẾT

- 1/ Máy trạng thái
- 2/ Các thiết bị

III. XÂY DỰNG MÔ HÌNH

- 1/ Máy trạng thái của mô hình
- 2/ Lưu đồ giải thuật
- 3/ Sơ đồ chân
- 4/ Hiện thực chương trình
- 5/ Lắp ráp mô hình

IV. ĐÁNH GIÁ

- 1/ Hiệu xuất mô hình
- 2/ Ưu điểm và nhược điểm

V. TỔNG KẾT

- 1/ Tóm tắt kết quả
- 2/ Hướng phát triển



I/ GIỚI THIỆU

1. Đặt vấn đề:

- Ở Việt Nam, các bãi xe được tự động hóa vẫn chưa đa dạng và có mặt ở nhiều thành thị. => Tiêu tốn nhân công không cần thiết => Chính vì thế đồ án này đưa ra một mô hình bãi đỗ xe thông minh có thể kiểm soát số lượng xe trong bãi đỗ và điều phối số lượng xe ra vào trong bãi đỗ một cách tự động giúp giảm lượng nhân công không cần thiết.

2. Mục tiêu

- Xây dựng được mô hình bãi đỗ xe tự động nhằm có thể ứng dụng phục vụ cho một nhóm người riêng biệt. Bãi đỗ xe có các chức năng cơ bản như mở, đóng cổng ra vào, quản lý số lượng xe đang có trong bãi và thông báo khi bãi xe đã đầy chỗ.

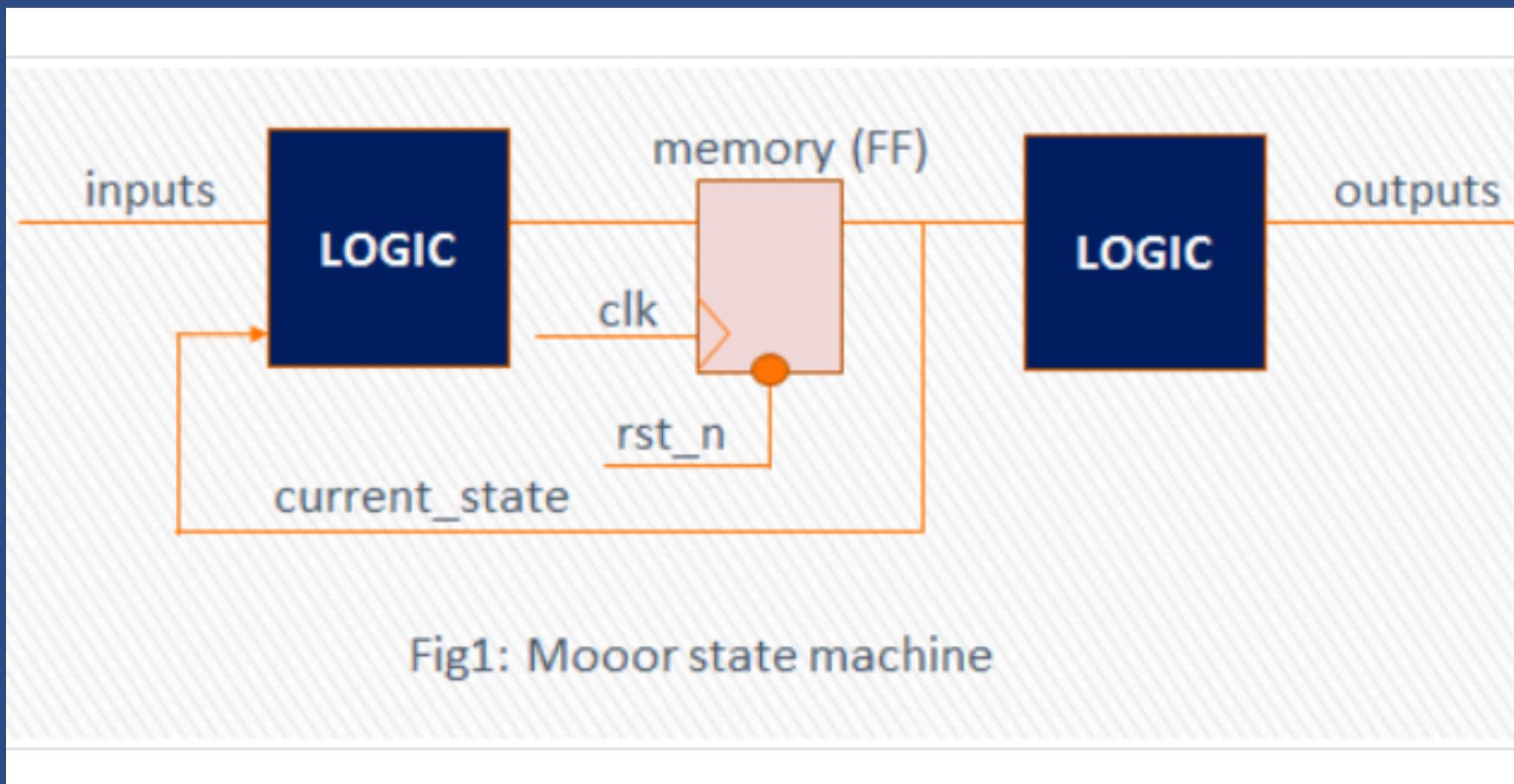
II. Cơ sở lý thuyết:

1. Máy trạng thái:

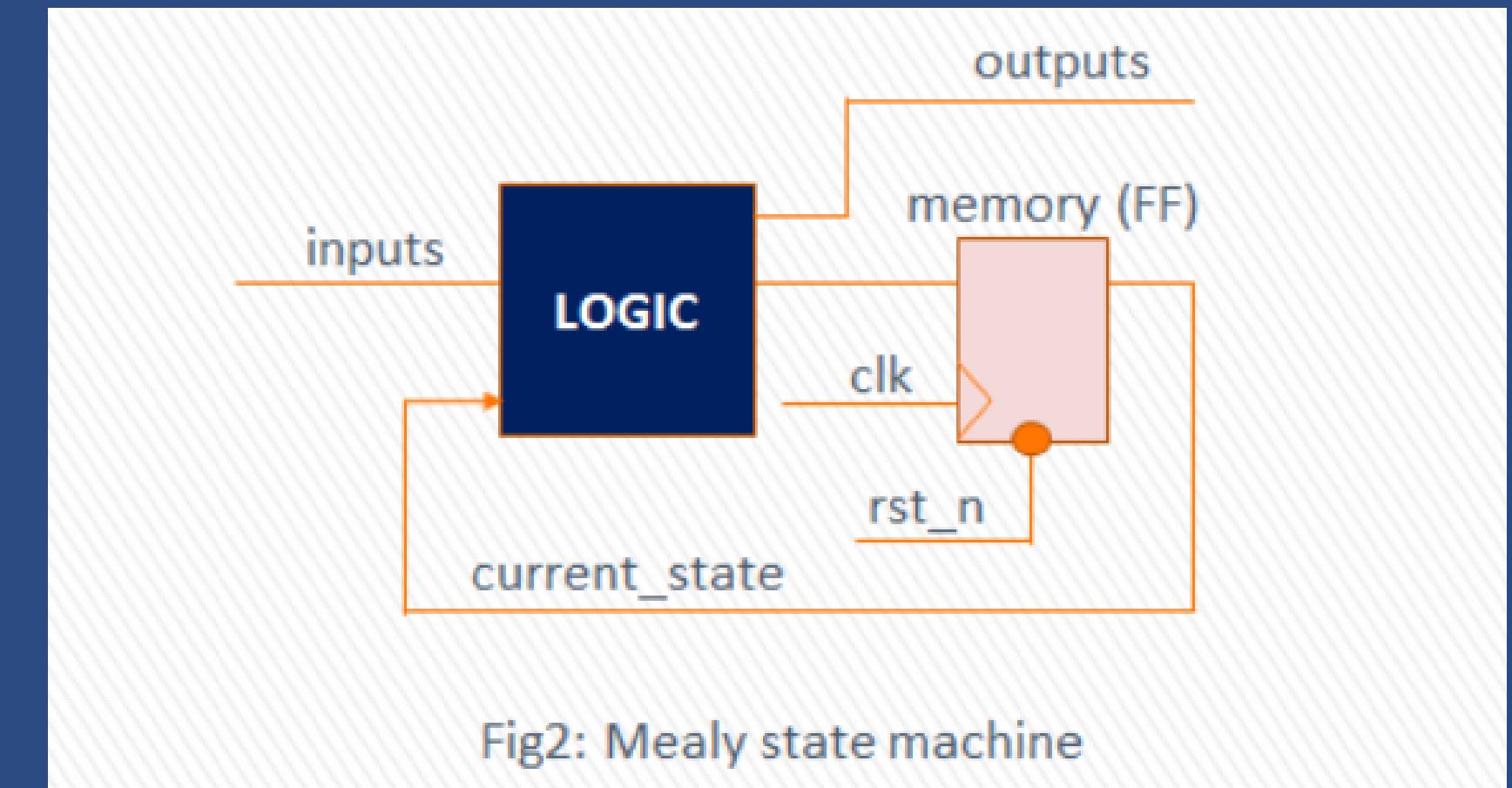
- Định nghĩa: Máy trạng thái là mô hình toán học dùng trong thiết kế máy tính và các mạch tuần tự. Máy trạng thái bao gồm nhiều trạng thái có liên quan với nhau, và tại một thời điểm, máy sẽ ở trong một trạng thái duy nhất. Trạng thái mà máy đang ở, trong bất kỳ thời điểm nào gọi là trạng thái hiện tại (current state), trạng thái mà máy sẽ chuyển đến kế tiếp từ thời điểm hiện tại gọi là trạng thái kế tiếp.
- Có 2 loại máy trạng thái:
 - +) Máy trạng thái Mealy (Mealy machine): Máy trạng thái Mealy là một máy trạng thái mà dữ liệu đầu ra được quyết định bởi trạng thái hiện tại và các dữ liệu ngõ vào.
 - +) Máy trạng thái Moor (Moor machine): Máy trạng thái Moor là máy trạng thái mà dữ liệu ngõ ra được quyết định duy nhất bởi trạng thái hiện tại.

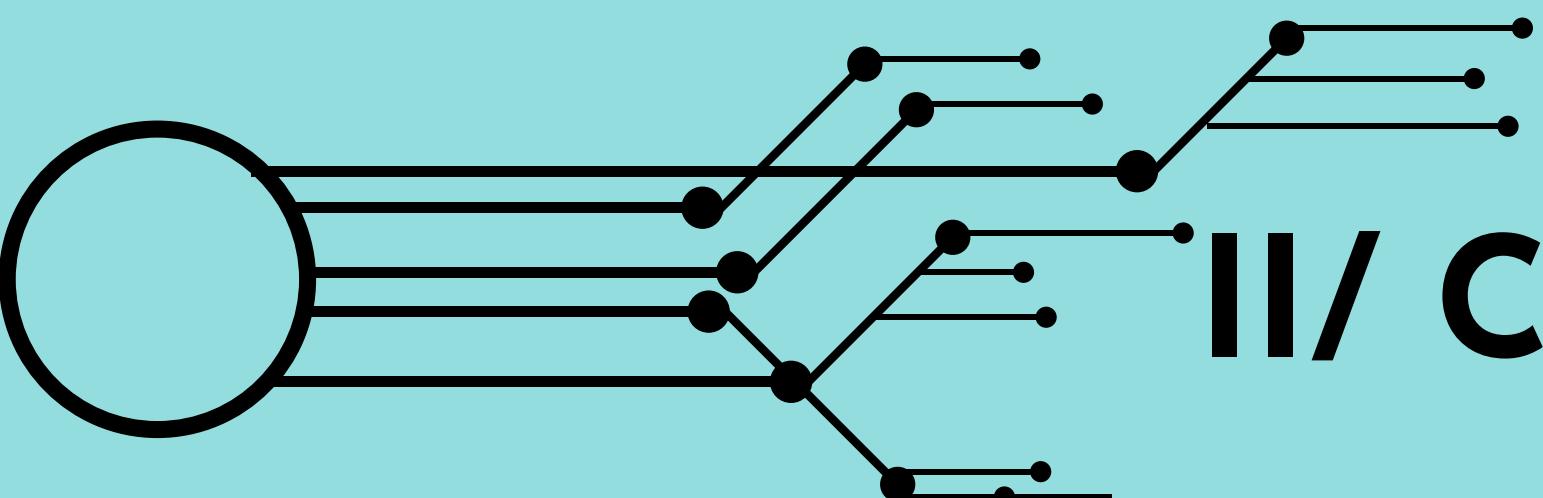
MÁY TRẠNG THÁI

Hình ảnh máy trạng thái kiểu Moore:



Hình ảnh máy trạng thái kiểu Mealy:





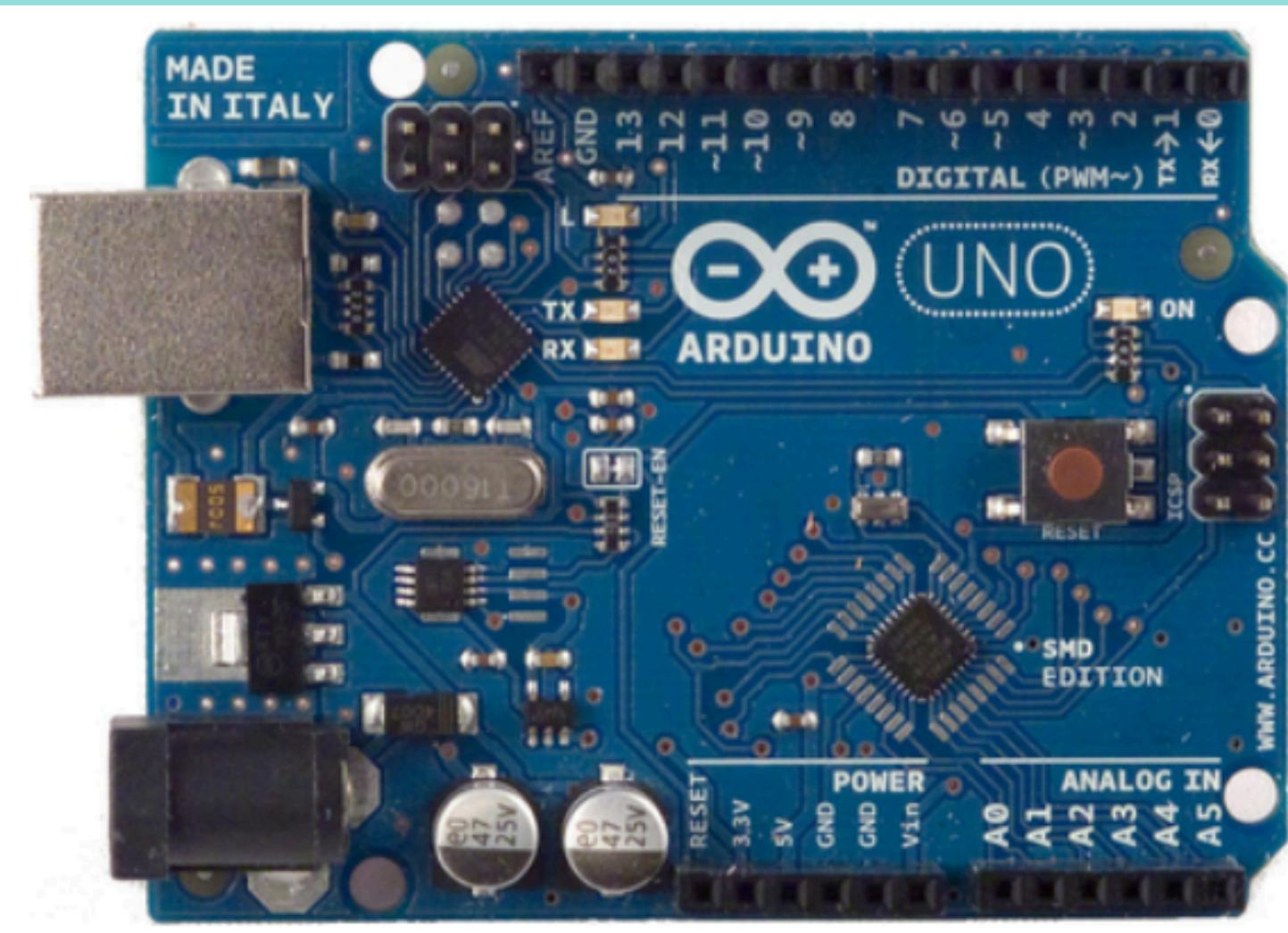
II/ CƠ SỞ LÝ THUYẾT

2. Các thiết bị:

- Vi điều khiển Arduino Uno R3

-Arduino Uno R3 là một bảng mạch vi điều khiển nguồn mở dựa trên vi điều khiển Microchip **ATmega328** được phát triển bởi Arduino.cc. Bảng mạch được trang bị các bộ chân đầu vào/ đầu ra Digital và Analog có thể giao tiếp với các bảng mạch mở rộng khác nhau.

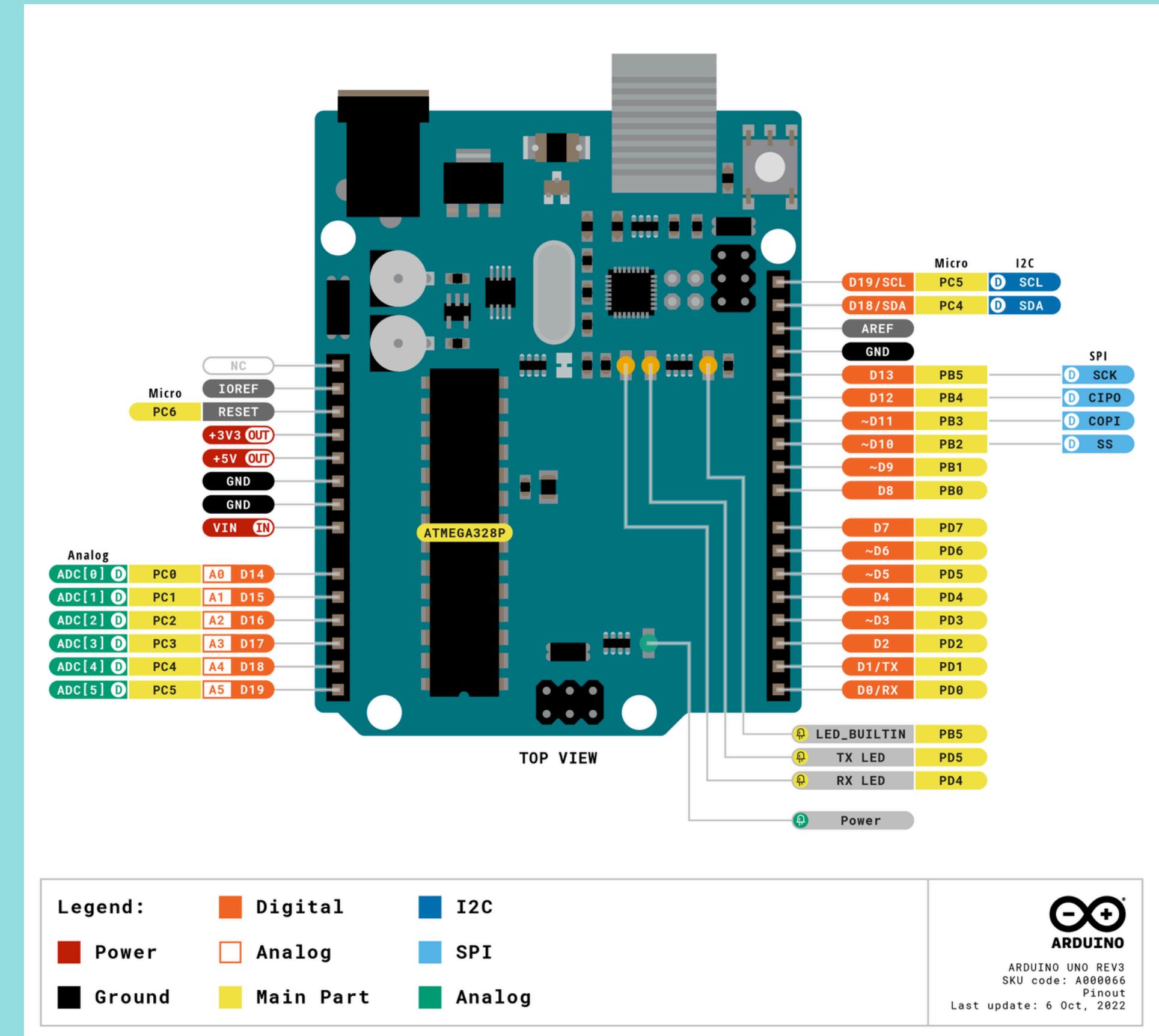
-Dựa trên nền tảng mở do Arduino.cc cung cấp các bạn dễ dàng xây dựng cho mình một dự án nhanh nhất (lập trình Robot, xe tự hành, điều khiển bật tắt led...).



2. Các thiết bị:

- Vi điều khiển Arduino Uno R3**

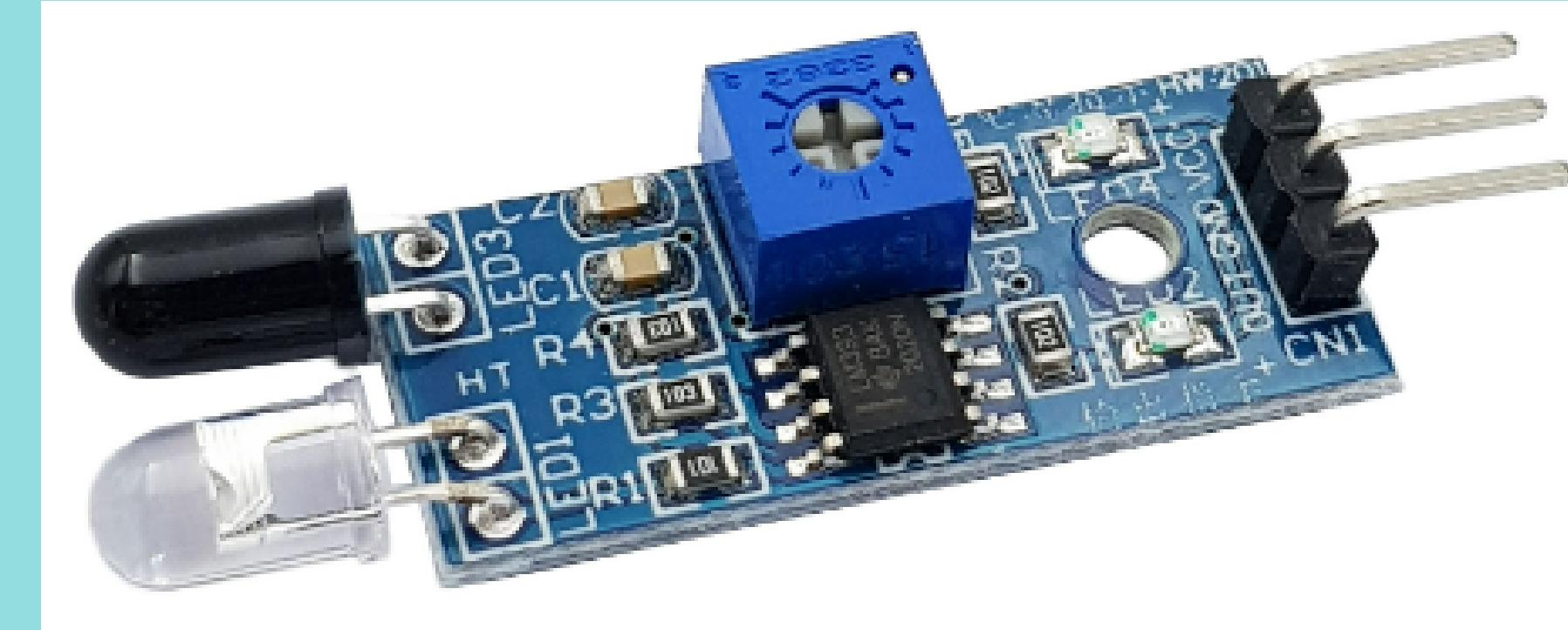
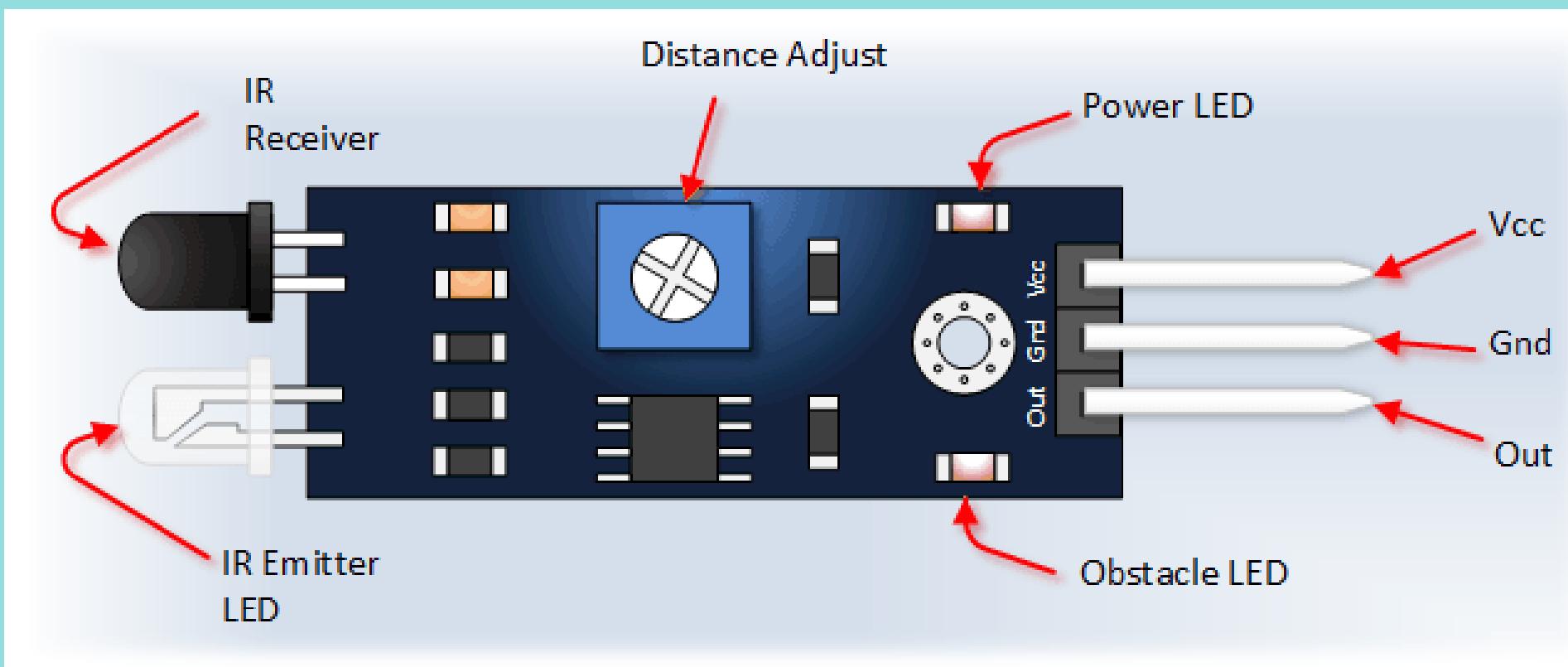
Thông số kỹ thuật	
Vì điều khiển	ATmega328
Điện áp hoạt động	5V(cấp qua cổng usb)
Điện áp khuyến nghị	6-9V
Số chân digital I/O	14 chân(6 chân PWM)
Số chân analog	6 chân
Dòng ra tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa(3.3V)	50 mA
Bộ nhớ Flash	32 KB (ATmega328) với 0.5 KB dùng bởi bootloader
SRAM	2 KB(ATmega328)
EEPROM	1 KB(ATmega328)
Giao động của thạch anh	16 MHz



2. Các thiết bị:

- Cảm biến vật cản hồng ngoại:

- Cảm biến vật cản hồng ngoại sử dụng một cặp truyền và nhận tia hồng ngoại. Tia hồng ngoại phát một tần số nhất định khi gặp vật cản sẽ phản xạ vào đèn thu hồng ngoại, sau khi qua IC so sánh đèn màu xanh sẽ sáng lên, đồng thời cho tín hiệu số đầu ra.
- Khoảng cách làm việc hiệu quả 2 ~ 5cm, điện áp làm việc là 3.3 V đến 5V. Độ nhạy sáng của cảm biến vật cản hồng ngoại được điều chỉnh bằng chíp, cảm biến dễ lắp ráp, dễ sử dụng,....
- Dùng rộng rãi trong robot tránh chướng ngại vật, xe tránh chướng ngại vật và dò đường.



2. Các thiết bị:

• Động cơ Servo SG90

-Động cơ Servo SG90 có kích thước nhỏ, là loại được sử dụng nhiều nhất để làm các mô hình nhỏ hoặc các cơ cấu kéo không cần đến lực nặng và có tốc độ phản ứng nhanh.

-Công dụng chính của động cơ servo là đạt được góc quay chính xác trong khoảng từ 90 độ đến 180 độ. Việc điều khiển này có thể ứng dụng để lái robot, di chuyển các tay máy lên xuống,...



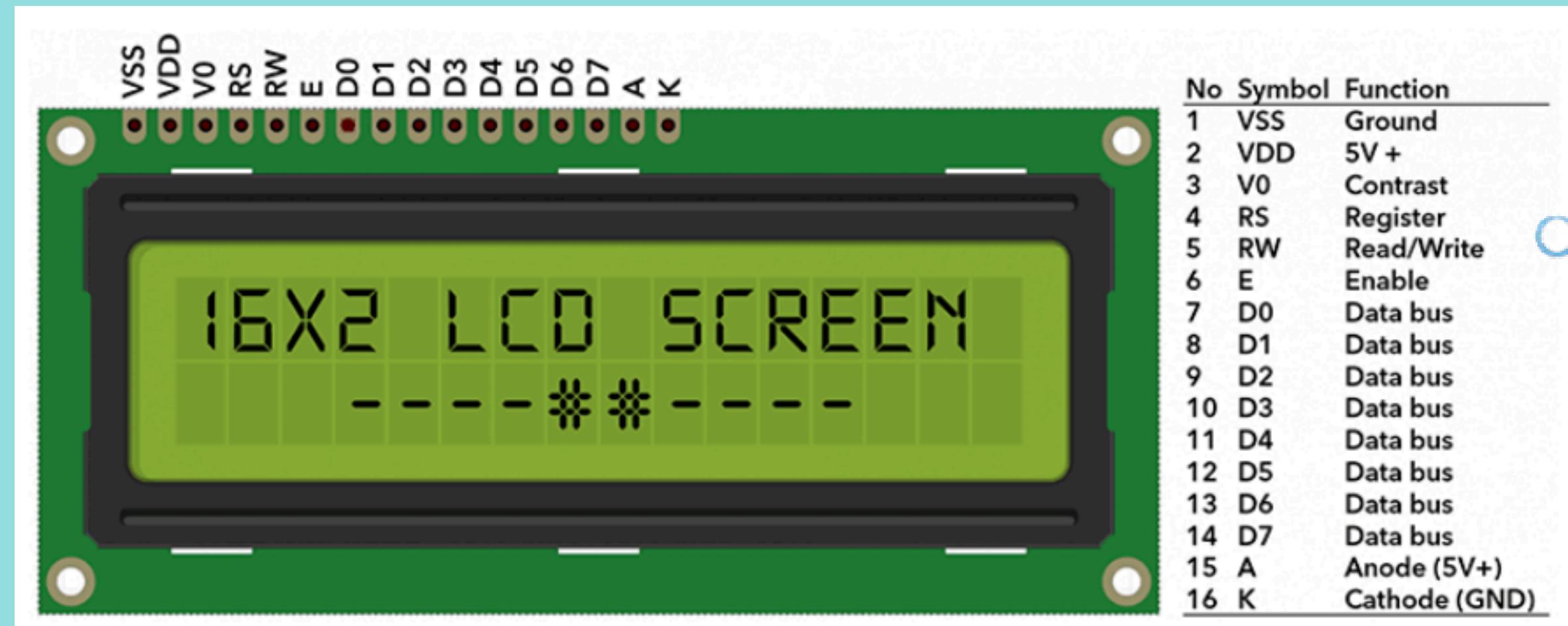
THÔNG SỐ KỸ THUẬT

- Điện áp hoạt động: 4.8-5VDC
- Tốc độ: 0.12 sec/ 60 deg (4.8VDC)
- Lực kéo: 1.6 Kg.cm
- Kích thước: 21x12x22mm
- Trọng lượng: 9g.

2. Các thiết bị:

• Màn hình LCD 1602

-Màn hình LCD 1602 sử dụng driver HD44780, có khả năng hiển thị 2 dòng với mỗi dòng 16 ký tự, màn hình có độ bền cao, rất phổ biến, nhiều code mẫu và dễ dàng sử dụng hơn khi sử dụng cùng với mạch chuyển tiếp I2C.



-Thông số kỹ thuật:

- Điện áp hoạt động: 5V
- Kích thước của màn hình LCD 1602: 8 x 3.6 x 0.8 cm
- Màu nền: xanh lá hoặc xanh dương
- Màu chữ: Màu đen

2. Các thiết bị:

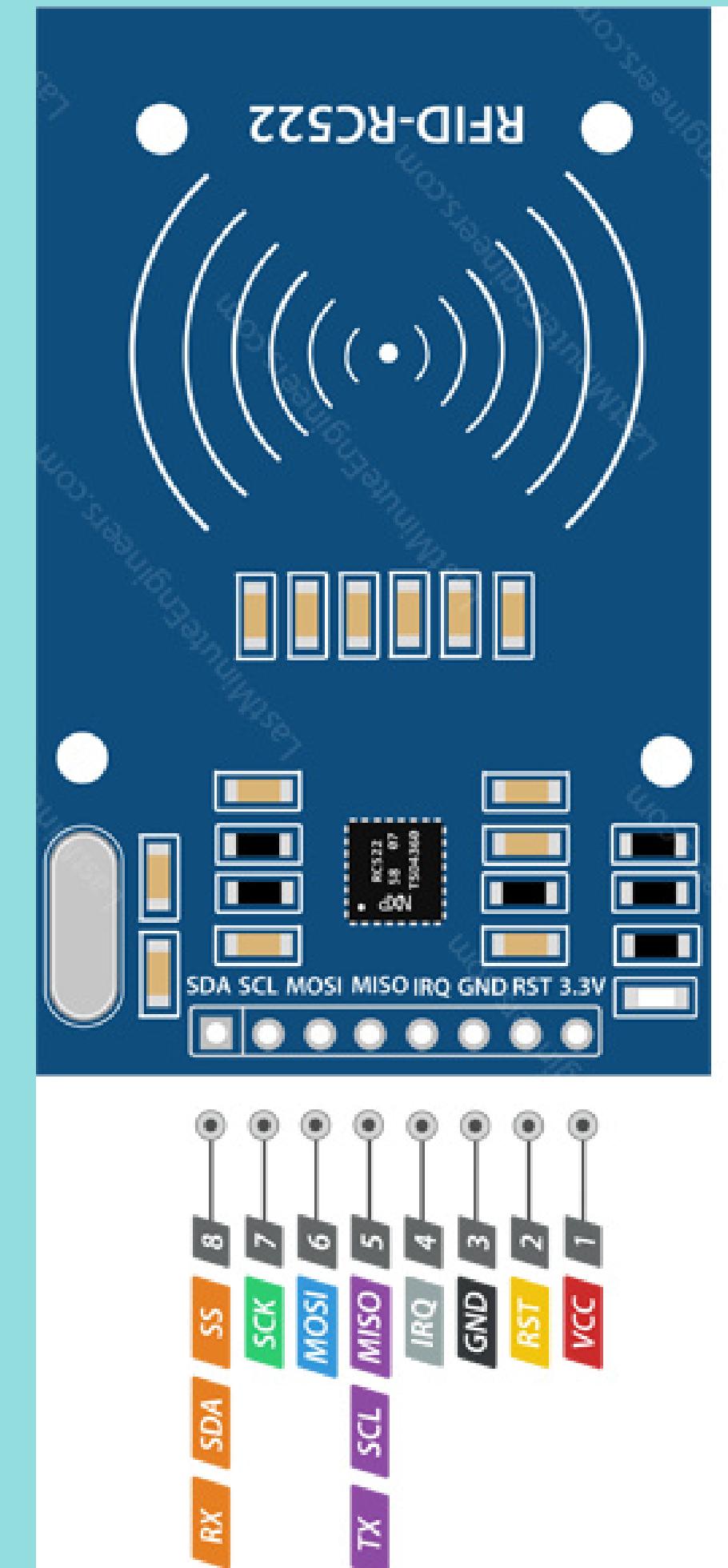
- RFID RC522 13.56MHz

-RFID là công nghệ nhận dạng đối tượng bằng sóng vô tuyến. Là một phương pháp nhận dạng tự động dựa trên việc lưu trữ dữ liệu từ xa, sử dụng thiết bị thẻ RFID và một Đầu đọc RFID.

-Module RFID RC522 dùng để đọc và ghi dữ liệu cho thẻ NFC tần số 13.56mhz được sử dụng quản lý nhà máy, quản lý chấm công, ...

-Thông số kỹ thuật

- Điện áp hoạt động: 3.3V
- Khoảng cách hoạt động: 0 - 60mm
- Tốc độ truyền dữ liệu: tối đa 10Mbit/s
- Nhiệt độ hoạt động: -20 đến 80 ° C
- Kích thước: 60mm×40mm





III/ XÂY DỰNG MÔ HÌNH

- 1. Máy trạng thái của mô hình**
- 2. Lưu đồ giải thuật**
- 3. Sơ đồ chân**
- 4. Hiện thực chương trình**
- 5. Lắp ráp mô hình**

1. Máy trạng thái của mô hình

a) Máy trạng thái của mô hình:

- Input: giá trị cảm biến vào (gtcambien1), giá trị cảm biến ra (gtcambien2), đọc thẻ thành công (ln1) , đọc đúng thẻ (ln2).
- Output là số lượng xe chứa trong bãi đỗ xe (đem).
- Máy trạng thái được hiện thực chỉ gồm một trạng thái.

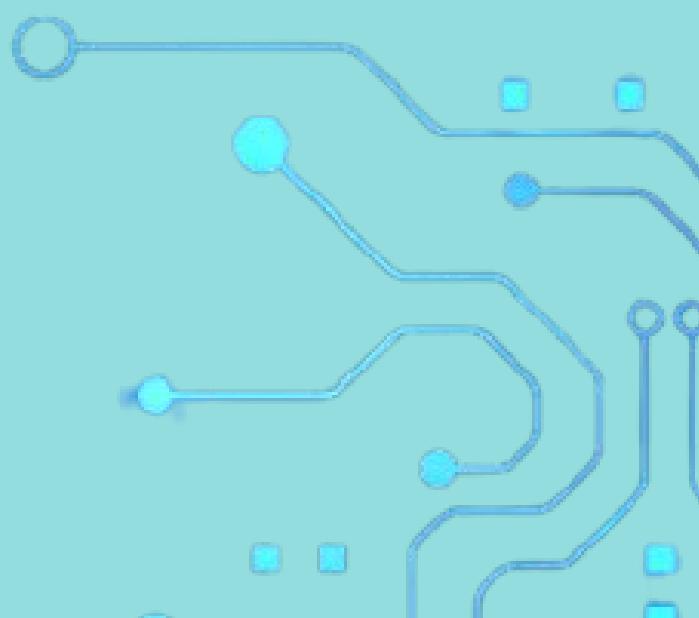
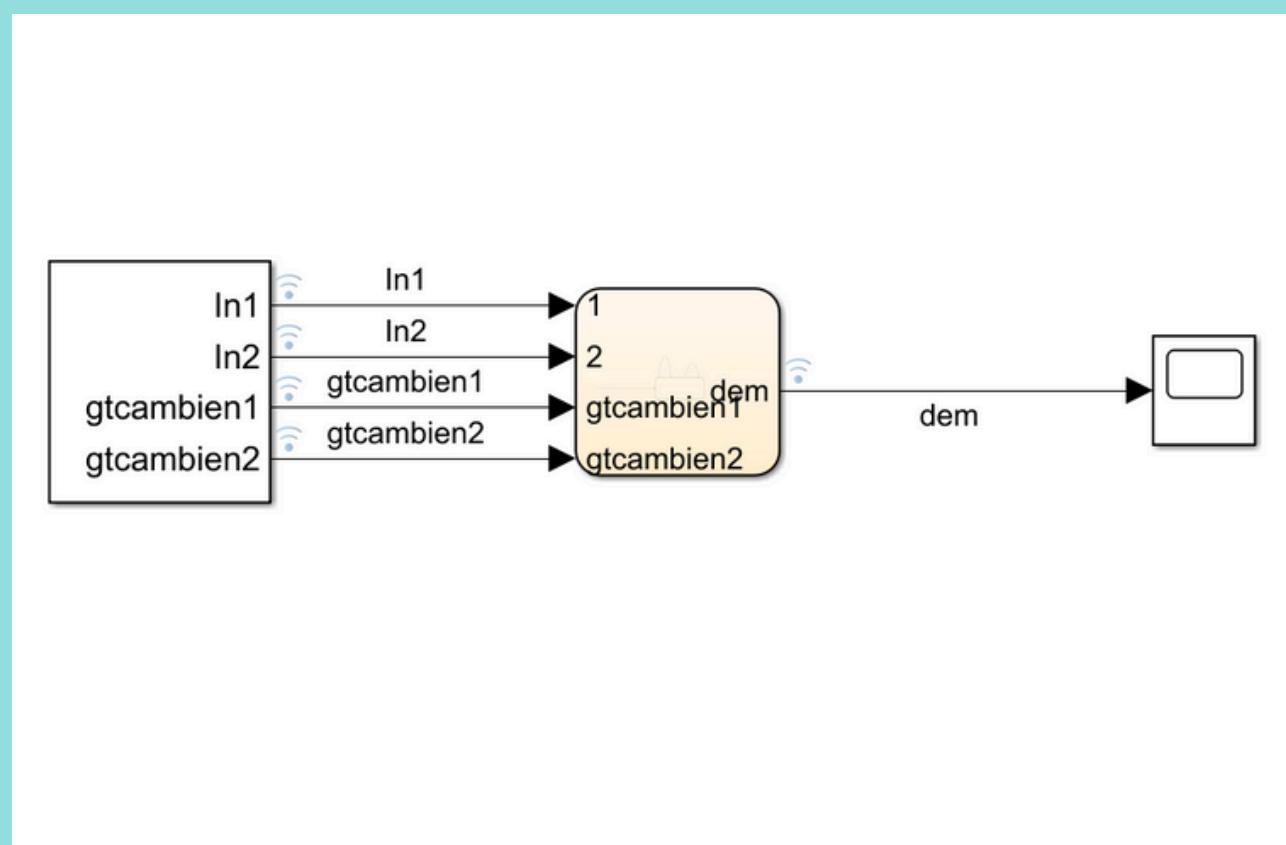
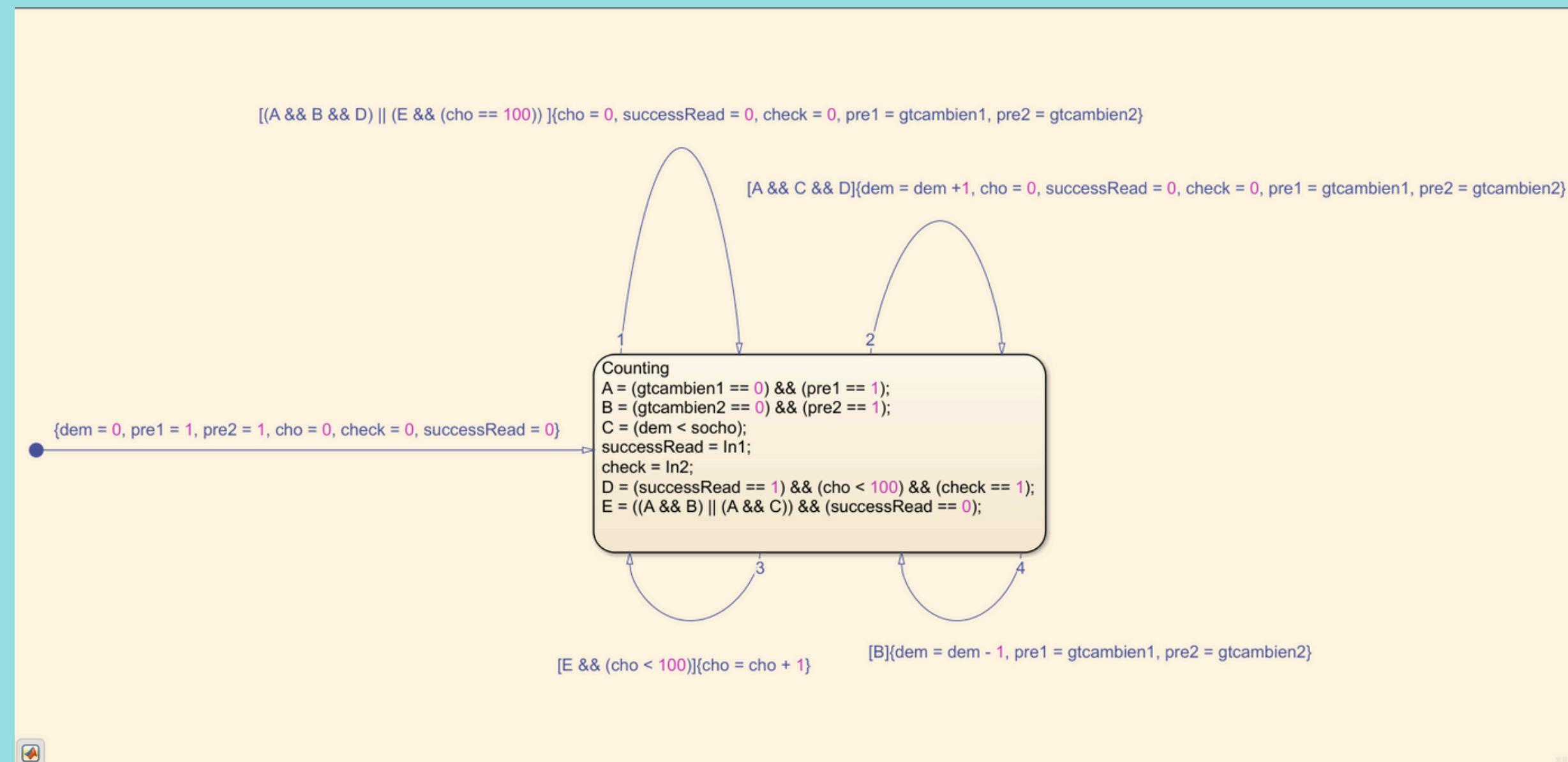
Ngoài các biến cục bộ được khai báo trong chương trình, trong máy trạng thái này ta còn dùng thêm một số biến điều kiện sau:

- A: điều kiện để có xe tới cổng vào,
- B: là điều kiện có xe tới cổng ra,
- C: là điều kiện kiểm tra xem bãi đỗ xe đã đầy xe chưa.
- D: là điều kiện nếu đọc thẻ thành công và thẻ đúng mà không quá thời gian chờ.
- E: là điều kiện có xe tới cổng vào nhưng đọc thẻ không thành công.

1. Máy trạng thái của mô hình

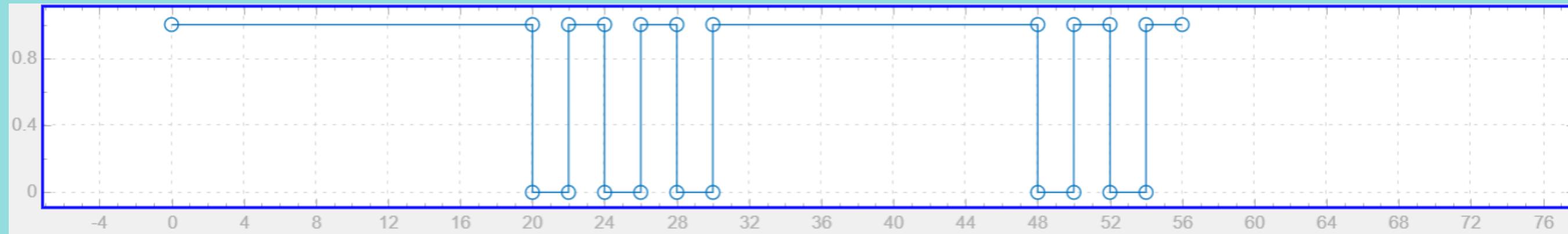
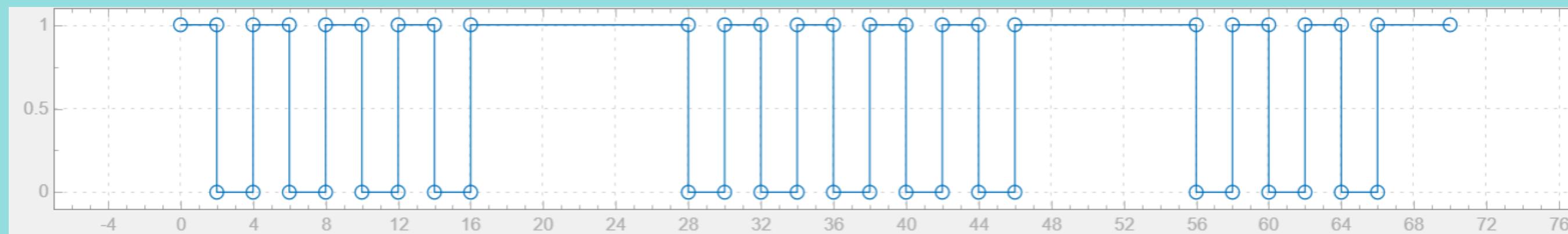
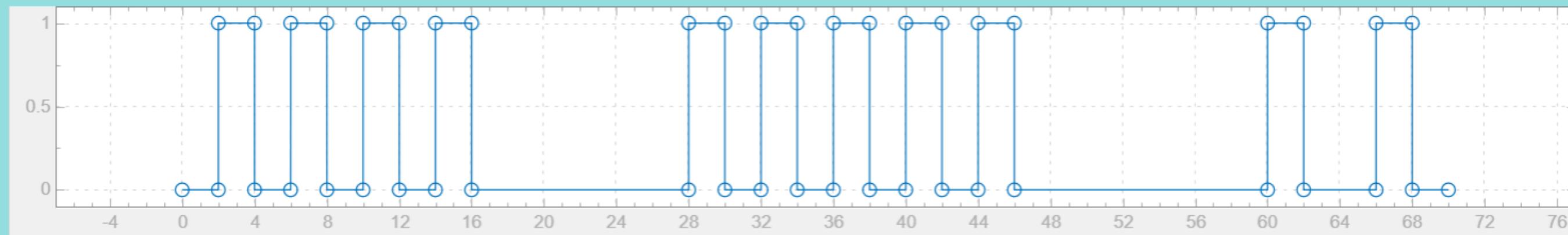
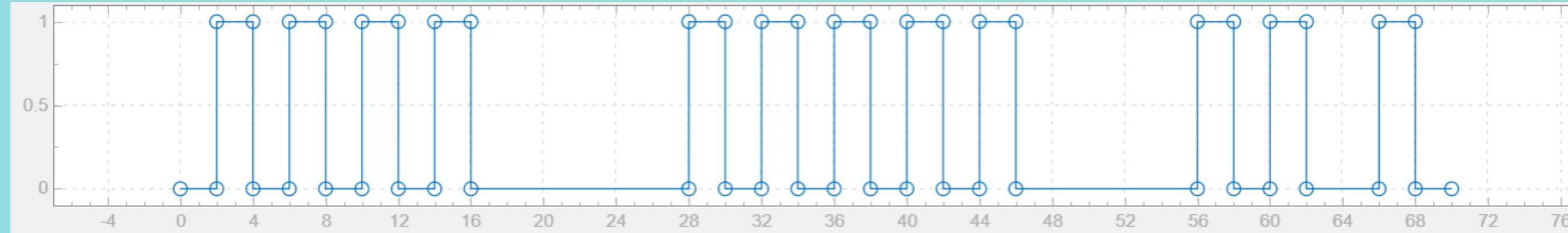
a) Máy trạng thái của mô hình:

Ta có máy trạng thái của mô hình được biểu diễn như sau:



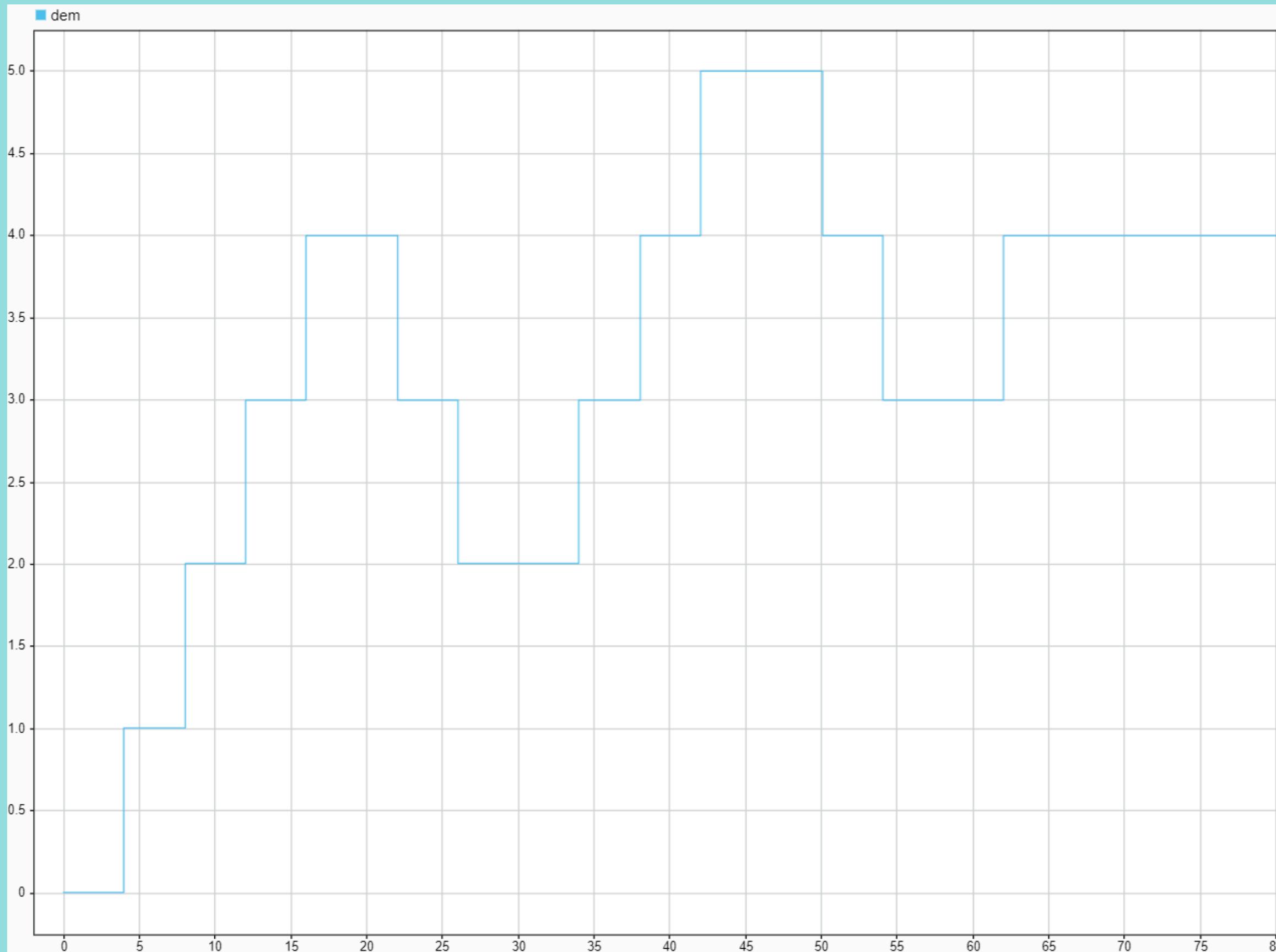
1. Máy trạng thái của mô hình

b) Mô phỏng chức năng: Ta thiết lập đầu vào dạng sóng cho 4 input sau:



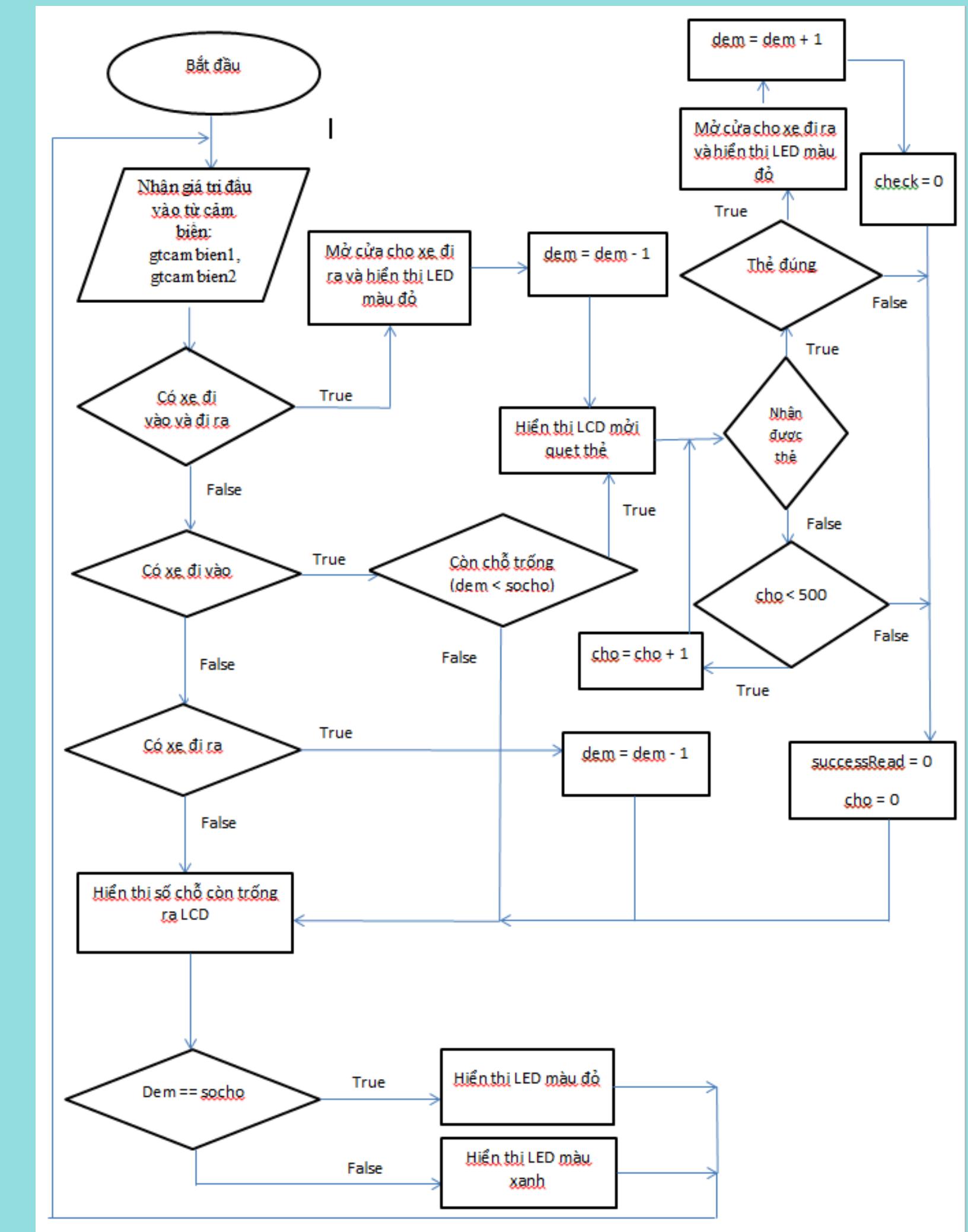
1. Máy trạng thái của mô hình

b) Mô phỏng chức năng: Ta có kết quả dạng sóng đầu ra như sau:



2. Sơ đồ giải thuật

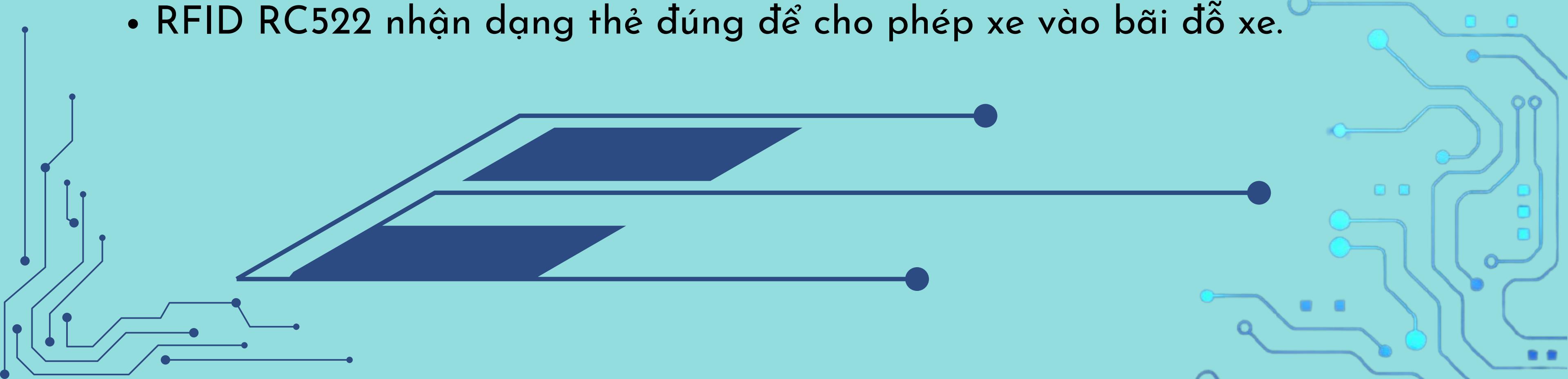
- Ta có sơ đồ giải thuật như hình bên.
- Để đơn giản, điều kiện có xe đi vào hay có xe đi ra hay kiểm tra thẻ được đọc và thẻ có đúng không ta thể hiện bằng lời. Tương tự như hiển thị LCD và hiển thị LED RGB. Chi tiết các điều kiện và các hành động hiển thị output sẽ được đề cập trong Phần IV.



3. Sơ đồ nối chân

a) Các Thiết bị sử dụng:

- Arduino UNO U3
- 2 cảm biến hồng ngoại có chức năng nhận biết xe vào/ra ở 2 cổng tương ứng.
- 2 động cơ servo SG90 điều khiển mở cổng vào/ra.
- LCD 1602 dùng để hiển thị số chỗ còn trống.
- LED RGB biểu thị trạng thái của bãi đỗ xe.
- RFID RC522 nhận dạng thẻ đúng để cho phép xe vào bãi đỗ xe.

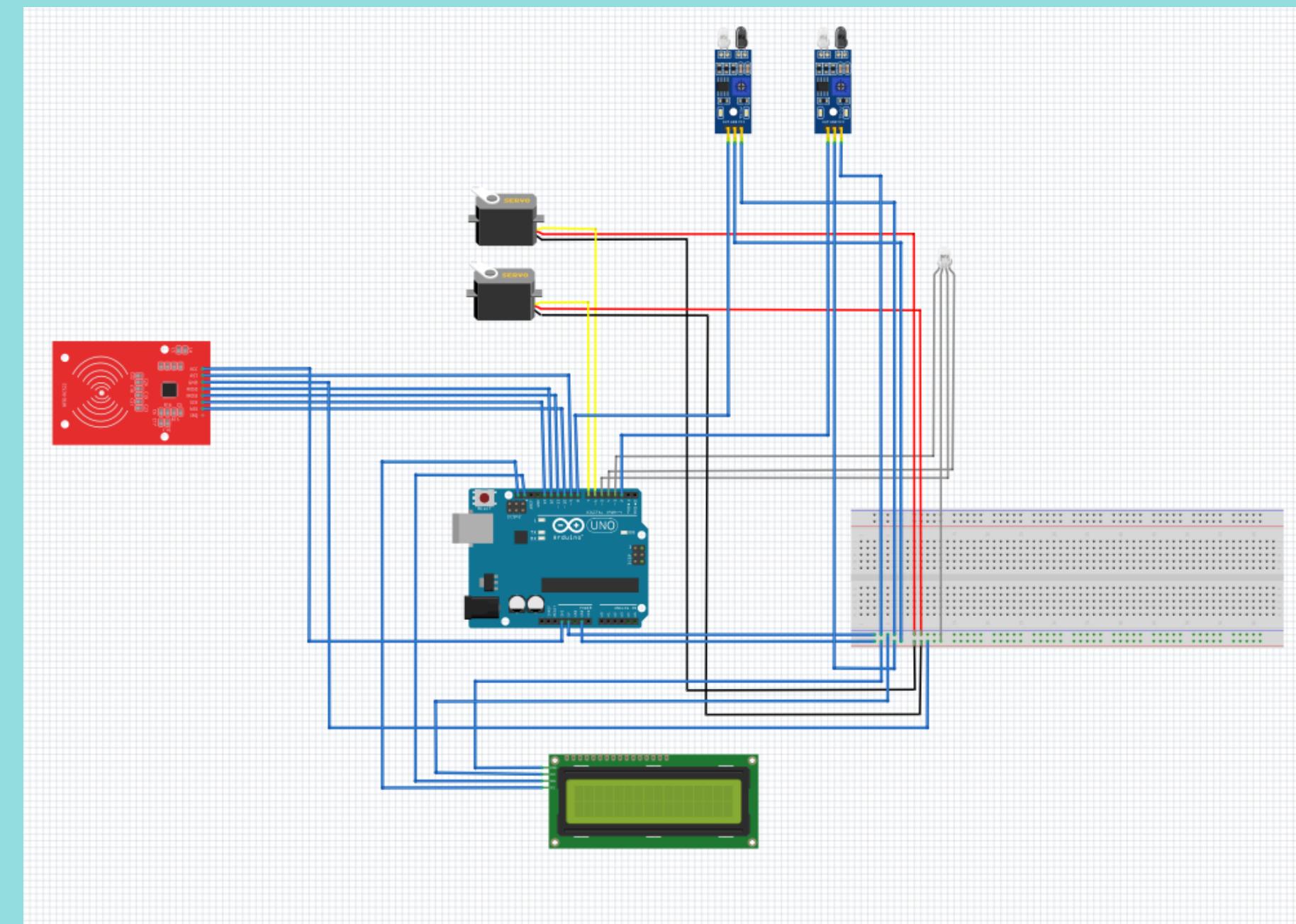


3. Sơ đồ nối chân

b) Sơ đồ nối chân của từng thiết bị trên:

Thiết bị	Tên chân	Chân dữ liệu trong arduino
Cảm biến hồng ngoại	Out (cảm biến xe đi vào)	D2
	Out (cảm biến xe đi ra)	D8
Động cơ servo	PWM (mở cổng đi vào)	D6
	PWM (mở cổng đi ra)	D7
LED RGB	REG	D3
	GREEN	D4
	BLUE	D5
LCD I2C	SDA	SDA (D18)
	SCL	SCL (D19)
RFID RC522	SS (SDA)	D10
	RST	D9
	MOSI	D11
	MISO	D12
	SCK	D13

c) Hình ảnh sơ đồ nối chân:



4. Hiện thực chương trình:

- Full Source Code: https://drive.google.com/file/d/1HnmJ5bEdK2rfxbY1-qDNscEWU9s6Dcr2/view?usp=drive_link

giải thích chi tiết chương trình: Khai báo các thư viện sử dụng, các chân và các module sử dụng

Đầu tiên, ta khai báo các thư viện, giá trị các chân sử dụng để kết nối giữa các cảm biến với arduino và các biến sử dụng trong chương trình. Ta hiện thực qua đoạn code sau:

```
// khai tao cac thu vien su dung
#include <SPI.h>
#include <MFRC522.h>
// hai thu vien tren dung cho viec doc the
#include <Wire.h>
#include <Servo.h> // thu vien giao tiep voi servo
#include <LiquidCrystal_I2C.h> // thu vien giao tiep voi LCD

// Khai bao cac chan
int cambien1 = 8, cambien2 = 2; // 2 chan cam bien vat can
#define RST_PIN          9      // chan rst cua MRC522
#define SS_PIN           10      // chan SDA cua MRC522
const int RED_PIN = 3;        // led đỏ
const int GREEN_PIN = 4;      // led xanh lá
const int BLUE_PIN = 5;       // led xanh dương

LiquidCrystal_I2C lcd(0x3F,16,2); // khai tao LCD 16x2
Servo myServo1, myServo2; // Khai bao servo
MFRC522 mfrc522(SS_PIN, RST_PIN); // khai bao module MRC522
```

4. Hiện thực chương trình:

Đầu tiên, ta khai báo các thư viện, giá trị các chân sử dụng để kết nối giữa các cảm biến với arduino và các biến sử dụng trong chương trình. Ta hiện thực qua đoạn code sau:

Khai báo các biến sử dụng trong chương trình

```
// Khai bao cac bien su dung trong chuong trinh
int UID[4], i;
// the trang: 229 135 163 172
// the xanh: 141 56 255 49
int ID[4] = {229, 135, 163, 172}; // UID cua Thẻ nhận dạng tín hiệu
int socho = 5, dem = 0, pre1 = 1, pre2 = 1, cho = 0; // so cho toi da la 5
bool check = 0, successRead = 0;
```

4. Hiện thực chương trình:

Một số hàm cơ bản điều khiển các thiết bị:

a) Các hàm điều khiển servo:

Ta điều khiển bằng cách thiết lập các góc quay cho servo để mở/đóng cổng vào/ra. Giữa mỗi lần quay, ta delay 1 khoảng thời gian để cho xe kịp đi vào/ra

- Hàm điều khiển servo mở/đóng cổng vào

```
void sservo1() { // mo cong vao  
    myServo1.write(180);  
    delay(1000);  
    myServo1.write(90);  
}
```

- Hàm điều khiển servo mở/đóng cổng ra

```
void sservo2() { // mo cong ra  
    myServo2.write(0);  
    delay(1000);  
    myServo2.write(90);  
    //delay(3000);  
}
```

4. Hiện thực chương trình:

Một số hàm cơ bản điều khiển các thiết bị:

b) Hàm đọc thẻ dùng RFID RC522:

Hàm này nhận diện thẻ đầu vào, đồng thời đọc ID của thẻ và so sánh với ID hợp pháp. Những thẻ có ID hợp pháp sẽ được đỗ xe trong bãi đỗ. Giá trị trả về của hàm là kiểu boolean. Hàm trả về là 0 nếu không có thẻ được đọc, ngược lại hàm trả về 1 khi có thẻ được đọc.

```
// ham doc the - xuuat ra serial monitor
bool docthe()
{
    if ( ! mfrc522.PICC_IsNewCardPresent() ) // kiem tra xem co phai the moi de doc khong
    {
        return;
    }
    if ( ! mfrc522.PICC_ReadCardSerial() ) // doc thong tin trong the
    {
        return;
    }

    Serial.print("UID của thẻ: ");

    for (byte i = 0; i < mfrc522.uid.size; i++)
    {
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        UID[i] = mfrc522.uid.uidByte[i];
        Serial.print(UID[i]);
    }

    Serial.println("  ");
    mfrc522.PICC_HaltA(); // ngung doc the lai
    mfrc522.PCD_StopCrypto1(); // dung MFRC522
    check = (UID[0] == ID[0]) && (UID[1] == ID[1]) && (UID[2] == ID[2]) && (UID[3] == ID[3]); // kiem tra xem UID co bang ID khong
    return 1;
}
```

4. Hiện thực chương trình:

Một số hàm cơ bản điều khiển các thiết bị:

c) Hàm đọc giá trị của cảm biến vật cản:

Hàm này đọc giá trị của cảm biến vật cản nhận biết xe vào/ra bãi đỗ, đồng thời tăng giảm số chỗ còn lại trong bãi đỗ xe. Khi một xe đi vào, số chỗ trống còn lại bị giảm đi 1, ngược lại khi một xe đi ra, số chỗ trống còn lại sẽ tăng lên một. Khi cảm biến vật cản nhận biết có một xe đang đi vào, nó sẽ chờ được đọc và xử lý bởi hàm đọc thẻ đã nêu trước đó rồi mới quyết định có mở cổng cho xe đi vào không.

```
void trangthai()
{
    int gtcambien1 = digitalRead(cambien1);
    int gtcambien2 = digitalRead(cambien2);
    if (gtcambien1 == 0 && pre1 == 1 && gtcambien2 == 0 && pre2 == 1)
    {
        sservo2();
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Moi quet the!");
        while (!successRead)
        {
            successRead = docthe();
            if (cho == 500) break;
            cho = cho + 1;
        }
        if (check)
        {
            display_led_R();
            sservol();
            delay(100);
            check = 0;
        }
        successRead = 0;
        cho = 0;
    }
}
```

```
else if(gtcambien1 == 0 && pre1 == 1 && dem < socho)
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Moi quet the!");
    while (!successRead)
    {
        successRead = docthe();
        if (cho == 500) break;
        cho = cho + 1;
    }
    if (check)
    {
        display_led_R();
        dem = dem + 1;
        sservol();
        delay(200);
        check = 0;
    }
    successRead = 0;
    cho = 0;
}
else if(gtcambien2 == 0 && pre2 == 1)
{
    display_led_R();
    dem = dem - 1;
    sservo2();
    delay(200);
}
pre1 = gtcambien1;
pre2 = gtcambien2;
UID[0] = 0;
UID[1] = 0;
UID[2] = 0;
UID[3] = 0;
```

4. Hiện thực chương trình:

Một số hàm cơ bản điều khiển các thiết bị:

d) Hàm output các giá trị ra màn hình LCD 1602:

Ta dùng hàm print_output() để xuất output ra màn hình LCD. Nếu trường hợp bãi đỗ xe không còn chỗ chứa, ta xuất ra màn hình LCD dòng chữ: “Đã hết chỗ chứa”. Còn nếu không, ta dùng hàm printLCD() để xuất ra số chỗ trống còn lại trong bãi đỗ xe và dòng chữ: “Xin mời vào” cho các xe vào đỗ bãi giữ xe.

Hàm print_output().

```
void print_output()
{
    if (dem == socho) {
        lcd.setCursor(0,0);
        lcd.print("Da het cho chua");
        lcd.setCursor(0,1);
        lcd.print("      ");
    }
    else printLCD();
}
```

Hàm printLCD().

```
void printLCD() {
    lcd.setCursor(0,0);
    lcd.print("So cho trong: ");
    lcd.print(socho - dem);
    lcd.setCursor(0,1);
    lcd.print("Xin moi vao");
}
```

4. Hiện thực chương trình:

Một số hàm cơ bản điều khiển các thiết bị:

e) Hàm hiển thị màu cho LED RGB:

-LED màu xanh lục khi bãi đỗ xe còn chỗ. LED chuyển sang màu đỏ khi có xe đang ra/vào bãi đỗ xe hoặc bãi đỗ xe đã hết chỗ chứa.

Hàm hiển thị màu đỏ cho LED RGB

```
void display_led_R() // sang den mau do
{
    digitalWrite(RED_PIN, HIGH);
    digitalWrite(GREEN_PIN, LOW);
    digitalWrite(BLUE_PIN, LOW);
}
```

Hàm hiển thị màu xanh lục cho LED RGB

```
void display_led_G() // sang den mau xanh la
{
    digitalWrite(RED_PIN, LOW);
    digitalWrite(GREEN_PIN, HIGH);
    digitalWrite(BLUE_PIN, LOW);
}
```

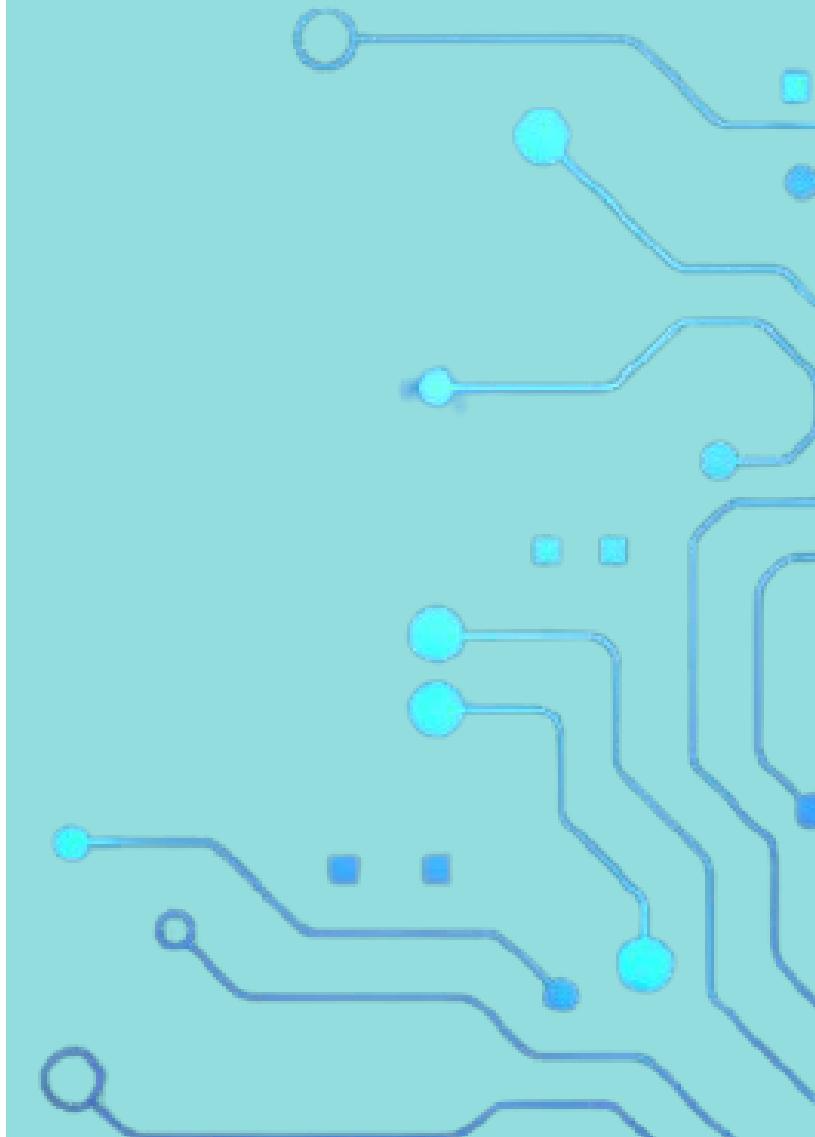
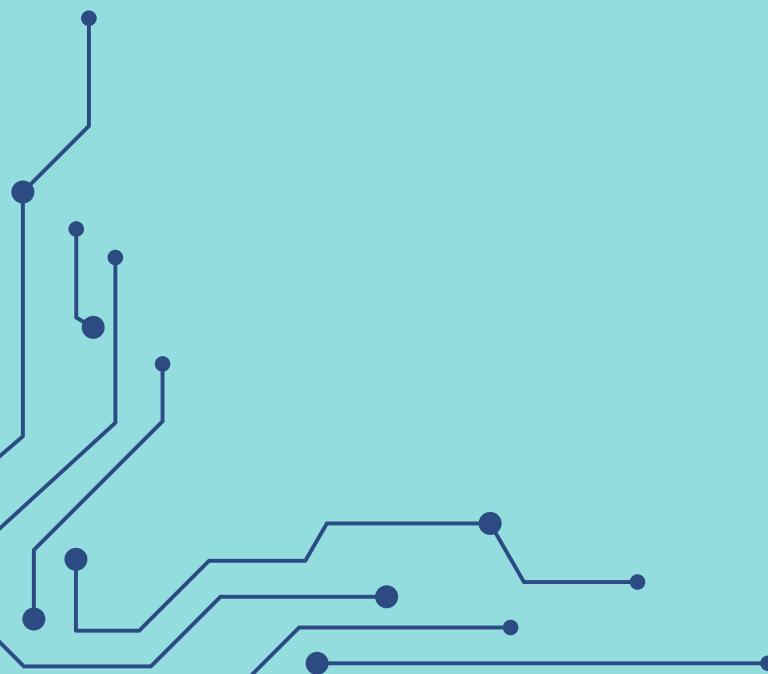
4. Hiện thực chương trình:

Một số hàm cơ bản điều khiển các thiết bị:

f) Hàm khởi tạo các tín hiệu và các thiết bị:

Trong hàm setup(), ta khởi tạo các tín hiệu cho 2 cảm biến vật cản, 2 servo, LCD 1602 và bật đèn nền của LCD. Khởi tạo tín hiệu cho các chân LED RGB, màn hình serial, giao thức SPI và MFRC522

```
// ham khai tao cac tin hieu
void setup() {
    // khai tao tin hieu 2 cam bien
    pinMode(cambien1, INPUT);
    pinMode(cambien2, INPUT);
    myServo1.attach(6); // chan servo vao
    myServo2.attach(7); // chan servo ra
    // khai tao vao bat den ngoai cho LCD
    lcd.init();
    lcd.backlight();
    // khai tao cac chan LED RGB
    pinMode(RED_PIN, OUTPUT);
    pinMode(GREEN_PIN, OUTPUT);
    pinMode(BLUE_PIN, OUTPUT);
    // khai dong man hinh serial
    Serial.begin(9600);
    // Khoi tao giao thuc SPI va module MFRC522
    SPI.begin();
    mfrc522.PCD_Init();
}
```



4. Hiện thực chương trình:

Một số hàm cơ bản điều khiển các thiết bị:

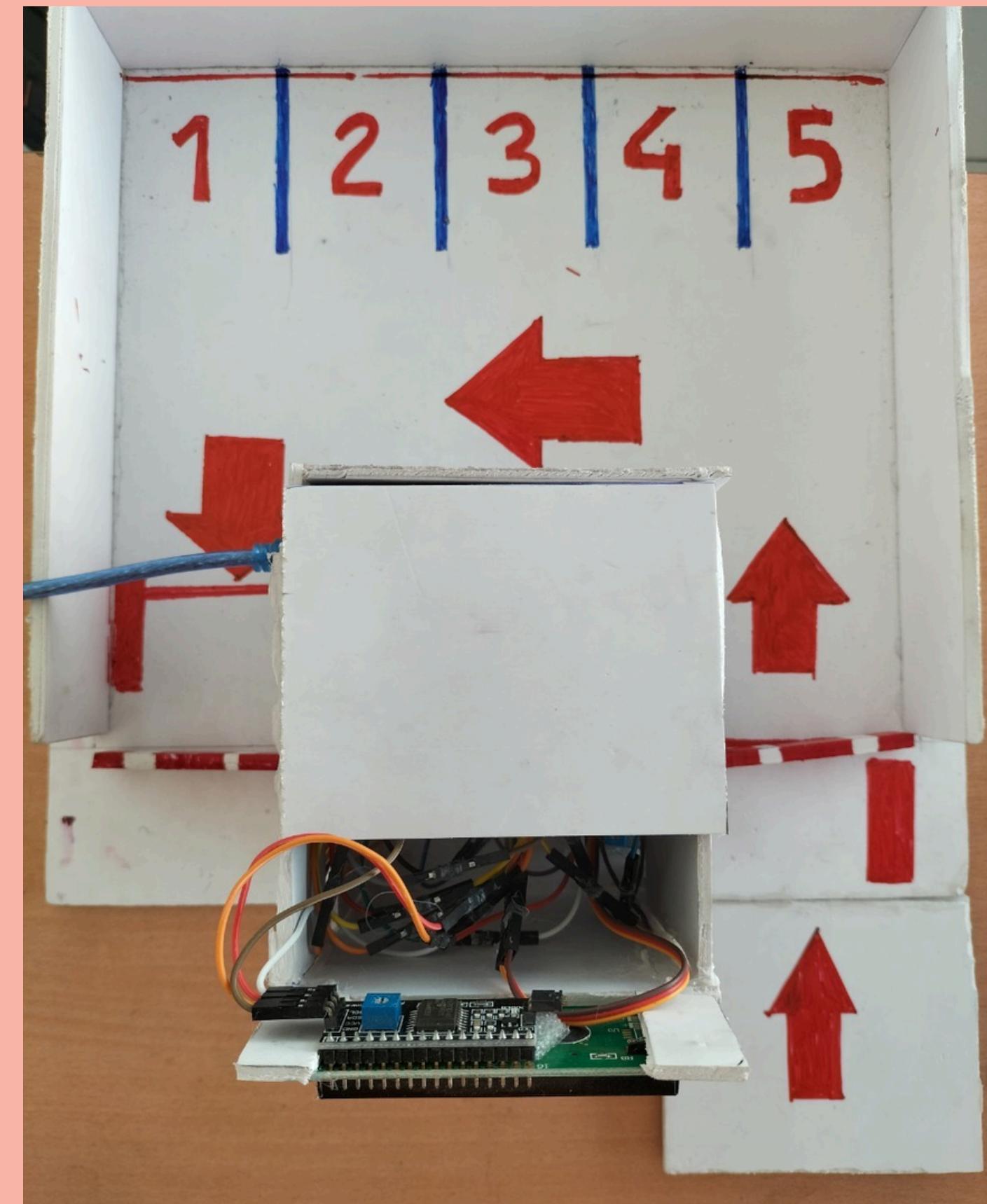
g) Hàm loop() lặp vô tận để luôn chạy chương trình:

Đầu tiên kiểm tra xem bãi đỗ xe đã đầy xe chưa, nếu rồi thì LED RGB hiển thị màu đỏ, còn nếu không thì LED RGB sẽ hiển thị màu xanh. Tiếp theo ta gọi hàm trangthai() là hàm đọc tín hiệu từ cảm biến vật cản để nhận biết xe vào/ra, sau đó gọi hàm print_output để xuất ra số chỗ trống còn lại trong bãi đỗ xe.

```
void loop() {  
    if (dem == socho) display_led_R(); // den do khi day cho  
    else display_led_G(); // nguoc lai den xanh  
    trangthai();  
    print_output();  
}
```

5. Lắp ráp mô hình:

Mô hình được hiện thực và lắp ráp ngoài đời:





IV. ĐÁNH GIÁ

1. Hiệu suất mô hình:

-Video chạy mô hình với mô tả tất cả các trường hợp xử lý mà mô hình có thể có:

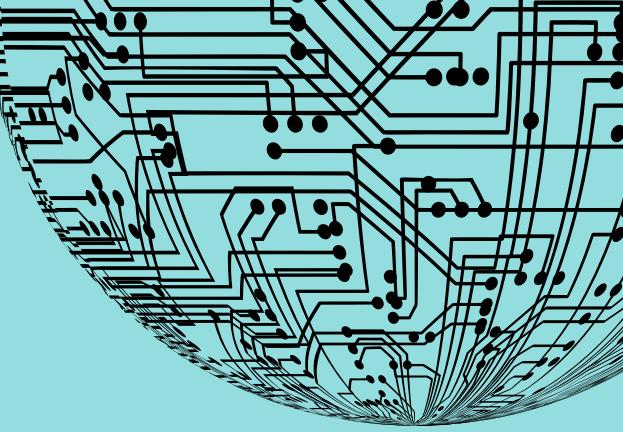
- Đường dẫn video chạy mô hình:

https://drive.google.com/file/d/12_yhgYb_eKsg-c1OsPQR8Hol8fufxcSh/view?usp=sharing

=> Dựa vào video chạy mô hình, ta nhận thấy đạt được kết quả chính xác với từng trường hợp đúng như mong đợi.



IV. ĐÁNH GIÁ



2. Ưu điểm và nhược điểm:

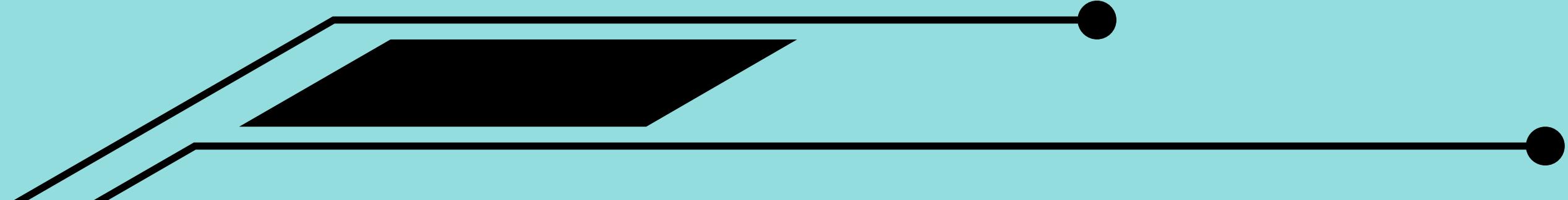
Ưu điểm:

- Các thành phần (cảm biến, Servo, LCD,...) được sử dụng để hiện thực mô hình là các thành phần có giá thành rẻ.
- Mô hình hoạt động hiệu quả và có độ chính xác cao, đúng như mong đợi.



Nhược điểm:

- Cảm biến có giá thành rẻ nên độ chính xác không cao dẫn đến sai số cho mô hình.
- Mô hình vẫn đôi lúc hoạt động không đúng, có sai số và có một số rẳng buộc nhất định mà người sử dụng phải tuân thủ.



V/ TỔNG KẾT

1. Tóm tắt kết quả:

- Chi phí xây dựng mô hình không đắt đỏ. Các thành phần để xây dựng mô hình là những thành phần có giá thành rẻ.
- Tuy chi phí xây dựng không đắt đỏ tuy nhiên mô hình xây dựng đã hoạt động hiệu quả và có độ chính xác cao.
- Mô hình có sai số tuy nhiên sai số không đáng kể và có thể chấp nhận được.
- Đây vẫn chỉ là mô hình, việc áp dụng thực tế để xây dựng một bãi đỗ xe thông minh với kích thước thật có thể gặp các khó khăn, trở ngại lớn hơn.



V/ TỔNG KẾT

2. Hướng phát triển tương lai:

- Có thể áp dụng để tiến hành hiện thực mô hình bãi đỗ xe thông minh vào đời sống với kích thước thật.
- Thay thế các cảm biến hiện tại bằng các cảm biến có độ sai số ít nhằm giảm thiểu sai số, nâng cao hiệu suất của mô hình.
- Có thể phát triển thêm một vài chức năng nâng cao khác mà một mô hình bãi đỗ xe thông minh có như còi báo cháy tự động, tự bật tắt đèn trong bãi đỗ xe dựa trên số lượng xe trong bãi đỗ xe,....
- Ứng dụng xử lý ảnh (nhận biết biển số xe) kết hợp với RFID nhằm nâng cao chức năng của mô hình.



Tài liệu tham khảo:

- <https://www.youtube.com/watch?v=gZ4hLL-SfdA&t=547s>
- <https://www.youtube.com/watch?v=2RNliDOwpN8&t=581s>
- <http://arduino.vn/bai-viet/181-gioi-thieu-servo-sg90-va-cach-dieu-khien-bang-bien-tro>
- <https://mecsu.vn/ho-tro-ky-thuat/giao-dien-lcd-i2c-voi-van-ban-chay-tren-man-hinh-arduino-vacac-ky-tu-tuy-chinh.QI5>
- <https://www.youtube.com/watch?v=GEv9e1pXjV4>
- <http://arduino.vn/bai-viet/42-arduino-uno-r3-la-gi>



Thank
you

