

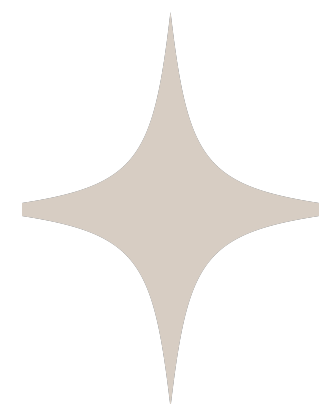
پروژه

ماشین هوشمند

2025

(SELF-DRIVING CAR)

فاطمه خیری

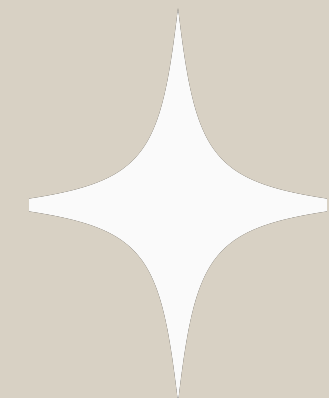


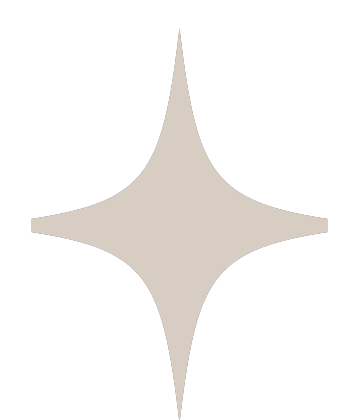
ماشین هوشمند (SELF-DRIVING CAR)

ACTION	PERCEPT SEQUENCE
MOVE FORWARD	[FORWARD, NO OBSTACLE]
STOP	[FORWARD, OBSTACLE DETECTED]
TURN RIGHT	[OBSTACLE ON LEFT, RIGHT CLEAR]
TURN LEFT	[OBSTACLE ON RIGHT, LEFT CLEAR]
STOP AND WAIT	[FORWARD, OBSTACLE CLOSE, BOTH SIDES BLOCKED]
TURN LEFT	[LEFT TURN AVAILABLE, NO OBSTACLE]
TURN RIGHT	[RIGHT TURN AVAILABLE, NO OBSTACLE]
MOVE FORWARD	[INTERSECTION, CLEAR PATH FORWARD]
CHOOSE CLEAR PATH (LEFT/RIGHT)	[INTERSECTION, OBSTACLE DETECTED AHEAD]
SLOW DOWN	[NARROW PATH AHEAD, OBSTACLE DETECTED]
SLOW DOWN AND TURN	[STEEP TURN, NO OBSTACLE]
RESUME MOVEMENT	[OBSTACLE CLEARED]
WAIT AND CHANGE LANE WHEN SAFE	[LANE CHANGE REQUIRED, OBSTACLE DETECTED]
STOP	[DESTINATION REACHED]

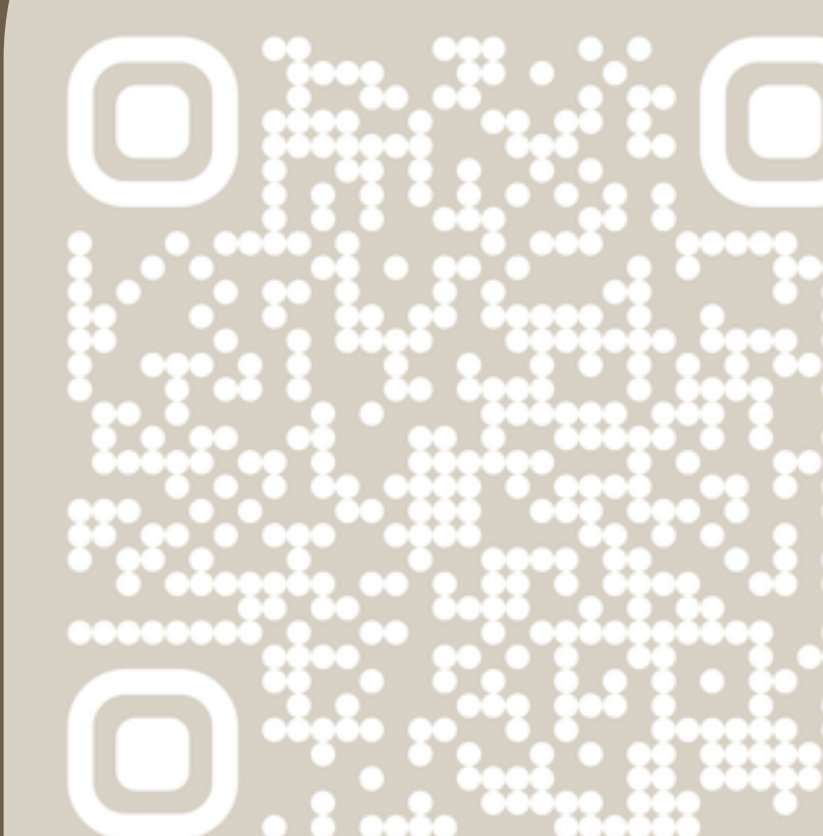
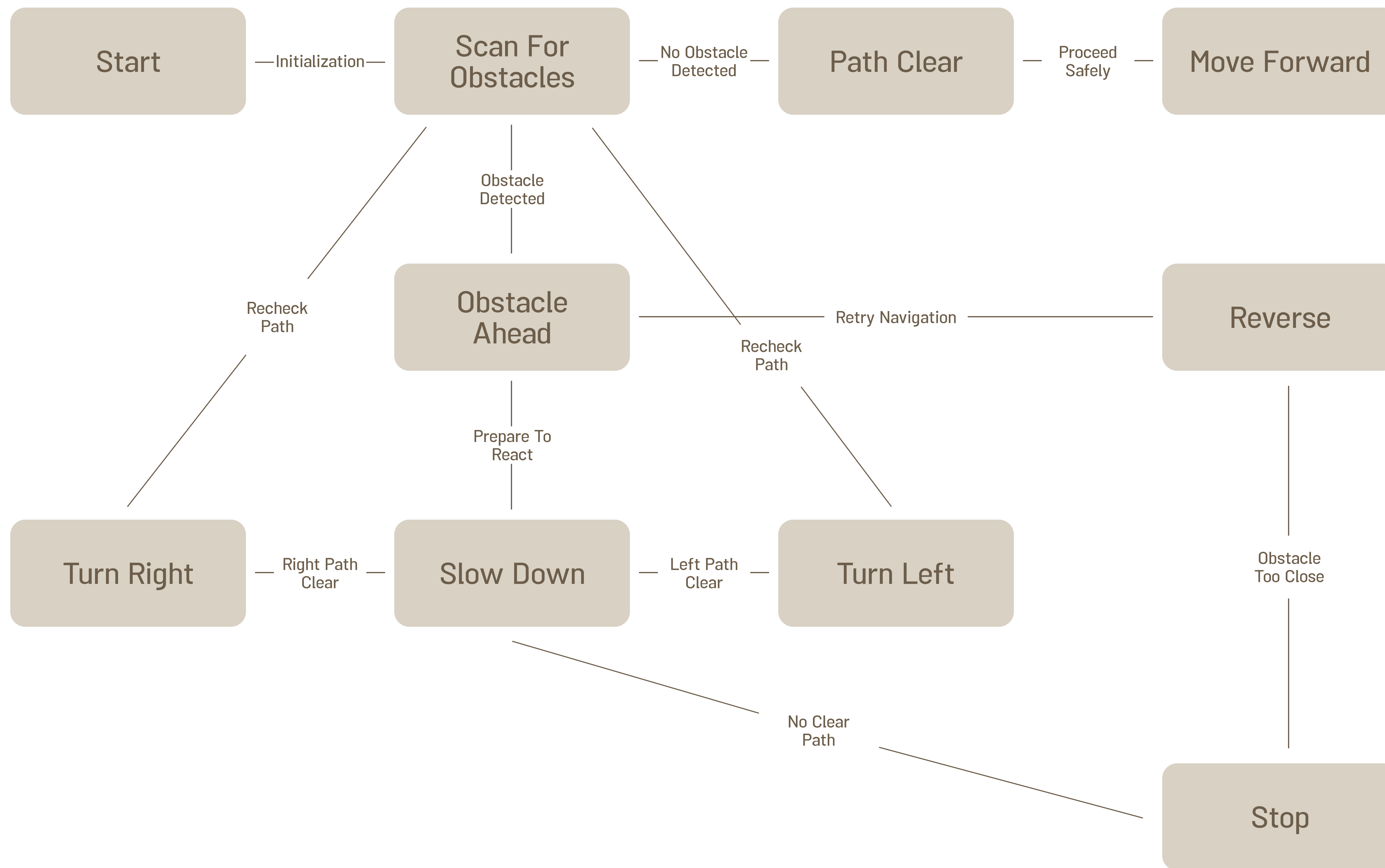


اسکن کنید

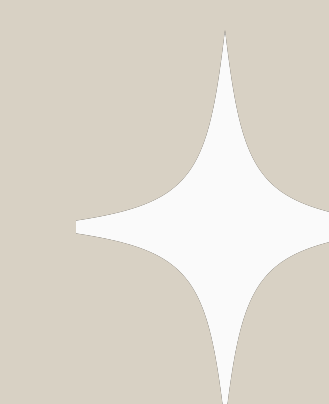


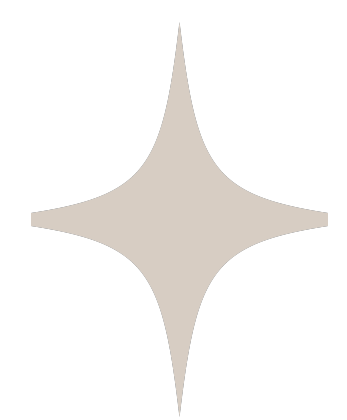


گراف ماشین هوشمند



اسکن کنید





گراف ماشین هوشمند

Start (شروع)

- ماشین از حالت اولیه شروع می‌کند. این مرحله شامل فرآیندهای راه‌اندازی سیستم‌ها مانند حسگرها و دوربین‌هاست.
- از حالت "Start" به "Scan for Obstacles" منتقل می‌شود تا جاده بررسی شود.

Slow Down (کاهش سرعت)

- در صورت تشخیص مانع، ماشین سرعت خود را کم می‌کند تا آماده واکنش شود.
- در این مرحله، ماشین بررسی می‌کند که آیا می‌تواند مسیر را تغییر دهد (چپ یا راست برود) یا نیاز است که متوقف شود.

Scan for Obstacles (اسکن برای موانع)

- ماشین از حسگرها و بینایی کامپیوتری برای بررسی مسیر و تشخیص موانع استفاده می‌کند.
- اگر مانعی پیدا نشود، به حالت "Path Clear" (مسیر خالی) می‌رود.
- اگر مانع شناسایی شود، به حالت "Slow Down" (کاهش سرعت) منتقل می‌شود.

Turn Left / Turn Right (چرخش به چپ یا راست)

- اگر حسگرها مسیر خالی در سمت چپ یا راست تشخیص دهند، ماشین به سمت مناسب تغییر جهت می‌دهد.
- پس از چرخش، دوباره مسیر را بررسی می‌کند.

Obstacle Ahead (مانع در مسیر)

- اگر مانع بسیار نزدیک باشد و امکان تغییر مسیر وجود نداشته باشد، ماشین به حالت "Stop" می‌رود.

Stop (توقف)

- ماشین برای جلوگیری از برخورد متوقف می‌شود. در این حالت ممکن است منتظر بماند یا فرآیند جدیدی برای بررسی مسیر آغاز کند.

Reverse (حرکت به عقب)

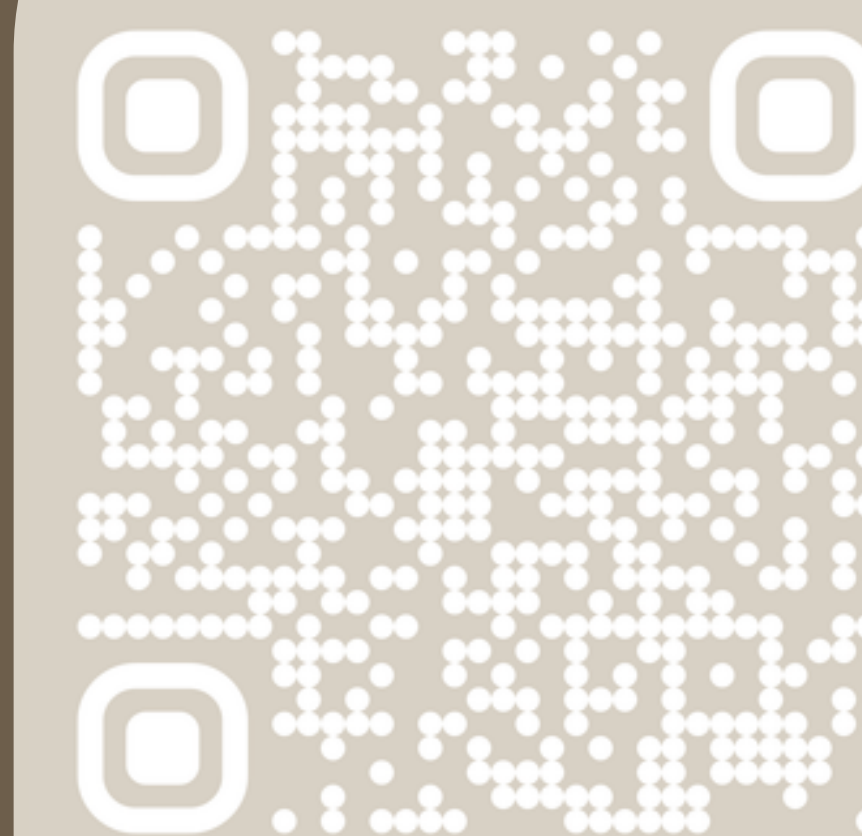
- اگر مانع بسیار نزدیک باشد و امکان چرخش یا حرکت به جلو وجود نداشته باشد، ماشین می‌تواند کمی به عقب برود و مسیر جدیدی را بررسی کند.

Path Clear (مسیر خالی)

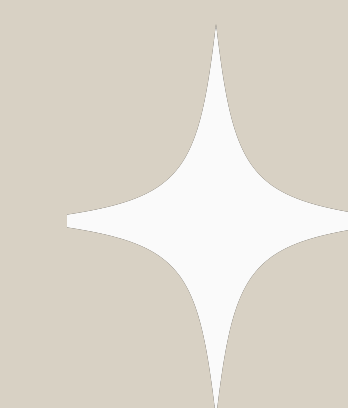
- اگر هیچ مانعی تشخیص داده نشود، ماشین به حالت "Move Forward" (حرکت به جلو) می‌رود.
- این نشان‌دهنده شرایط ایمن برای ادامه مسیر است.

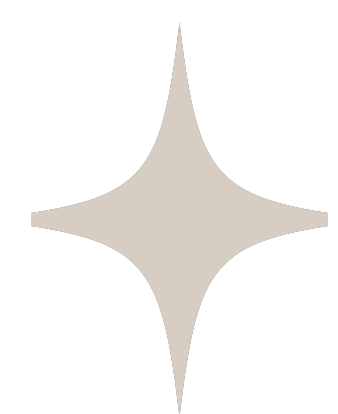
Move Forward (حرکت به جلو)

- ماشین در این حالت با سرعت مناسب و با نظارت دائمی حرکت می‌کند.
- هر لحظه امکان بازگشت به "Scan for Obstacles" برای بررسی مسیر وجود دارد.



اسکن کنید





کد ماشین هوشمند

```
import time

# حالات ماشین
states = [
    "Start",
    "Scan for Obstacles",
    "Move Forward",
    "Turn Left",
    "Turn Right",
    "Slow Down",
    "Stop",
    "Reverse",
    "Path Clear",
    "Obstacle Ahead"
]

# توابع برای هر حالت
def start():
    print("Initializing system...")
    time.sleep(1)
    return "Scan for Obstacles"

def scan_for_obstacles():
    print("Scanning for obstacles...")
    time.sleep(1)
    # شبیه سازی نتیجه اسکن
    obstacle_detected = input("Obstacle detected? (yes/no): ").lower()
    if obstacle_detected == "no":
        return "Path Clear"
    else:
        return "Obstacle Ahead"

def move_forward():
    print("Path is clear. Moving forward...")
    time.sleep(1)
    return "Scan for Obstacles"

def turn_left():
    print("Obstacle detected. Turning left...")
    time.sleep(1)
    return "Scan for Obstacles"

def turn_right():
    print("Obstacle detected. Turning right...")
    time.sleep(1)
    return "Scan for Obstacles"

def slow_down():
    print("Narrow path ahead. Slowing down...")
    time.sleep(1)
    return "Scan for Obstacles"

def stop():
    print("Stopping the car...")
    time.sleep(1)
    return "Start"
```

```
def reverse():
    print("Reversing to avoid obstacle...")
    time.sleep(1)
    return "Scan for Obstacles"

def path_clear():
    print("Path clear. Proceeding safely...")
    time.sleep(1)
    return "Move Forward"

def obstacle_ahead():
    print("Obstacle detected ahead. Making a decision...")
    time.sleep(1)
    decision = input("Choose action (left/right/reverse/stop): ").lower()
    if decision == "left":
        return "Turn Left"
    elif decision == "right":
        return "Turn Right"
    elif decision == "reverse":
        return "Reverse"
    elif decision == "stop":
        return "Stop"
    else:
        print("Invalid decision. Stopping...")
        return "Stop"

# نگاشت وضعیت ها به توابع
state_functions = {
    "Start": start,
    "Scan for Obstacles": scan_for_obstacles,
    "Move Forward": move_forward,
    "Turn Left": turn_left,
    "Turn Right": turn_right,
    "Slow Down": slow_down,
    "Stop": stop,
    "Reverse": reverse,
    "Path Clear": path_clear,
    "Obstacle Ahead": obstacle_ahead
}

# اجرای شبیه سازی
def main():
    current_state = "Start"
    while True:
        print(f"Current State: {current_state}")
        current_state = state_functions[current_state]()

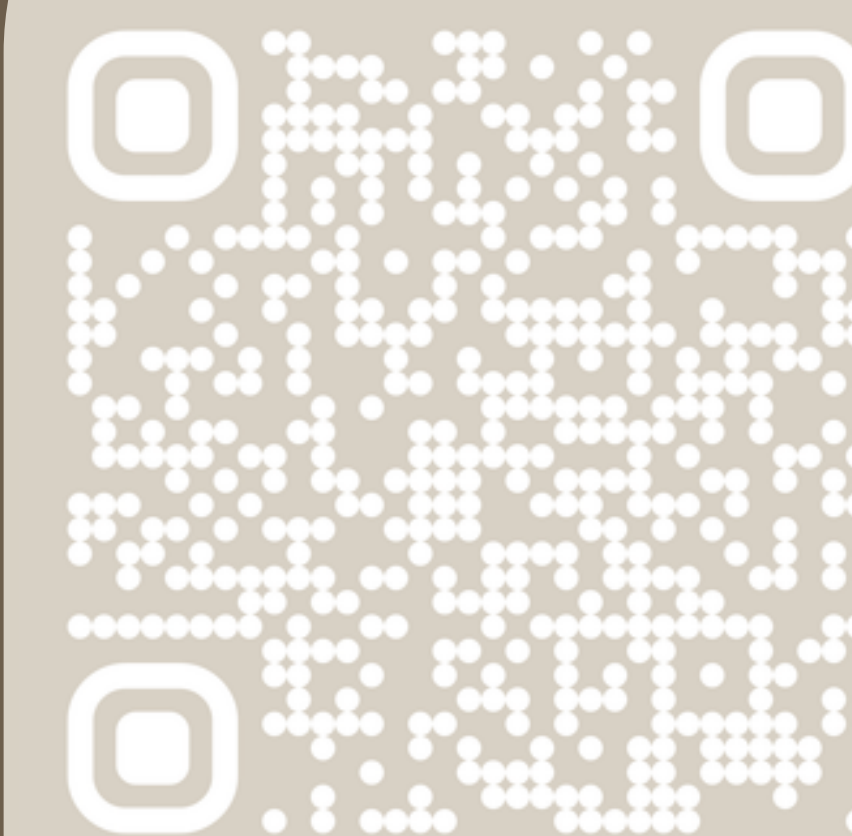
if __name__ == "__main__":
    main()
```

نحوه کارکرد:

- این کد یک سیستم ساده برای تغییر وضعیت ماشین هوشمند شبیه سازی می کند.
- ورودی ها به صورت دستی (کاربر) وارد می شوند تا وضعیت محیط مشخص شود.

بهبودهای ممکن:

- می توانید از داده های واقعی سنسور و مدل های یادگیری ماشینی (با استفاده از OpenCV و TensorFlow) برای تصمیم گیری خودکار استفاده کنید.
- با ترکیب جدول و نمودار، الگوریتم های پیشرفته تری پیاده سازی کنید.



اسکن کنید

