

```
# Glavna petlja koja stalno provjerava stanje senzora i odlučuje o ponašanju robota
while True:

    # Reset vrijednosti servo ugla i brzine pri svakom ciklusu

    servo_angle = 0

    speed = 1

    # Ažuriranje prethodnih vrijednosti rastojanja

    delta_dist_left = dist_left

    delta_dist_front = dist_front

    delta_dist_right = dist_right

    # Uzimanje novih vrijednosti sa ultrazvučnih senzora

    dist_left = TXT_M_I1_ultrasonic_distance_meter.get_distance()

    dist_right = TXT_M_I3_ultrasonic_distance_meter.get_distance()

    dist_front = TXT_M_I2_ultrasonic_distance_meter.get_distance()

    # Izračunavanje promjene u rastojanju

    delta_dist_left = dist_left - delta_dist_left

    delta_dist_front = dist_front - delta_dist_front

    delta_dist_right = dist_right - delta_dist_right

    # Ako je prepreka detektovana ispred i trenutno se ne skreće

    if (dist_front <= 45) and not turning and (start_turn_cd_time is None or time.time() -
start_turn_cd_time > TURN_CD):

        turning = True

        start_turn_time = time.time()
```

```

start_turn_cd_time = None

counter += 1

# Odabir pravca skretanja
if turn_orientation == "":
    if dist_left < dist_right:
        turn_orientation = "right"
    else:
        turn_orientation = "left"

# Ako je aktivan manevar skretanja
if turning:
    if turn_orientation == "right":
        servo_angle += 1 # Skretanje desno
    elif turn_orientation == "left":
        servo_angle -= 1 # Skretanje levo

# Provjerava se da li se robot može vratiti na pravolinijsko kretanje
if dist_front > 50 and time.time() - start_turn_time > TURN_DURATION:
    turning = False
    start_turn_time = 0
    start_turn_cd_time = time.time()
    turn_orientation = ""
else:
    # Korekcija ugla u zavisnosti od bočnih rastojanja

```

```
if turn_orientation == "right":
```

```
    servo_angle += 1.3 * (DESIRED_DIST - dist_left) / DESIRED_DIST
```

```
if turn_orientation == "left":
```

```
    servo_angle -= 1.3 * (DESIRED_DIST - dist_right) / DESIRED_DIST
```

```
# Bezbjednosna provera: ako je detektovano previše skretanja, robot se zaustavlja
```

```
if counter >= 12:
```

```
    if end_time == 0:
```

```
        end_time = time.time()
```

```
    if time.time() - end_time > TIMEOUT:
```

```
        break
```

```
# Ograničavanje servo ugla
```

```
servo_angle = min(1, max(-1, servo_angle))
```

```
# Slanje komandi za upravljanje motorima
```

```
set_servo(servo_angle)
```

```
set_speed(speed)
```

```
update_image()
```

```
# Zaustavljanje robota nakon izvršene misije
```

```
set_speed(0)
```