Alexandria University

Faculty of Engineering

Department of Electrical power and machines

Microprocessors

Final project

Line Follower Robot using 8051 microcontroller

Names	*ID*
عبد الرحمن احمد محمد جادو	21010716
زياد عادل حسين حسن	21010572

Line Follower Robot using 8051-microcontroller

Line follower Robot is a machine which follows a line, it may be a black line or a white line. Basically, two types of line follower robots are: one is black line follower which follows black line and second one is white line follower which follows white line. Line follower robot senses the line and run over it

- ## Components:

AT 89S51 microcontroller *[instead of 8051]*	*Crystal 12 MHZ*
Capacitors	***Resistances***
Dc motors	***Wheels and chassis***
12-volt DC battery	***IR sensors***
Push button	***L298 motor driver module***
Bread board	***Jumpers and wires***

The key components of the Line Follower Robot using the AT89S51 microcontroller and their roles in the project:

1. Chassis:

- ***The chassis serves as the physical frame for the robot, providing structural support and a platform to integrate other components.***
- ***A sturdy chassis ensures stability, durability, and ease of component attachment, contributing to the overall reliability of the robot.***

2. DC Motors:

- ***Geared DC motors drive the robot's wheels, enabling controlled movement and precise navigation.***
- ***The choice of motors influences the robot's speed, agility, and responsiveness to control signals. Geared motors are preferred for better torque control.***

3. Wheels:

- ***High-traction wheels facilitate smooth movement and traction, especially important for effective line tracking.***
- ***The selection of appropriate wheels impacts the robot's ability to follow the line accurately and navigate different terrains.***

4. Infrared Sensors:

- ***Infrared sensors are positioned on the underside of the robot to detect the line on the ground.***
- ***These sensors serve as the eyes of the robot, providing real-time feedback on the position of the line. Accurate and timely sensor data is crucial for effective line following.***

5. 8051 Microcontroller:

- ***The AT89S51 microcontroller acts as the brain of the robot, processing sensor data and making decisions based on a programmed algorithm.***
- ***Its versatile I/O capabilities and processing power make it well-suited for embedded systems. The microcontroller executes the line detection algorithm, controls motors, and orchestrates overall robot behavior.***

6. L298 motor driver module (H-Bridge):

- ***The motor driver, typically an H-bridge, is responsible for controlling the direction and speed of the DC motors.***
- ***H-bridge circuits enable bi-directional control of motors, allowing the robot to move forward, backward, turn, and adjust speed, all of which are essential for accurate line following.***

7. Power Supply:

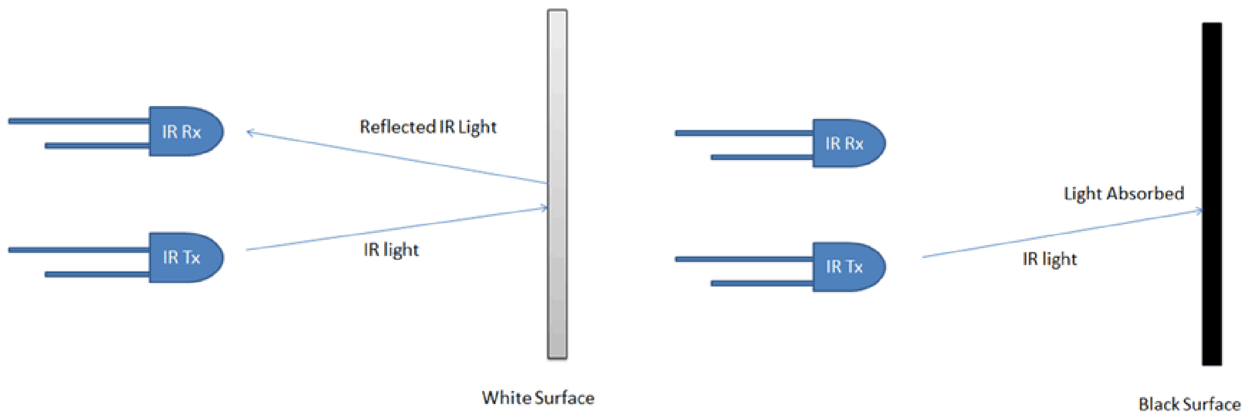
- ***The power supply provides the necessary electrical energy to operate the robot's components.***
- ***A stable and regulated power supply ensures consistent and reliable operation of the robot. Rechargeable batteries are commonly used to power the robot during its movements.***

8. Crystal Oscillator:

- ***provides 12MHz clock to the microcontroller.***

• Concept of Line Follower Robot:

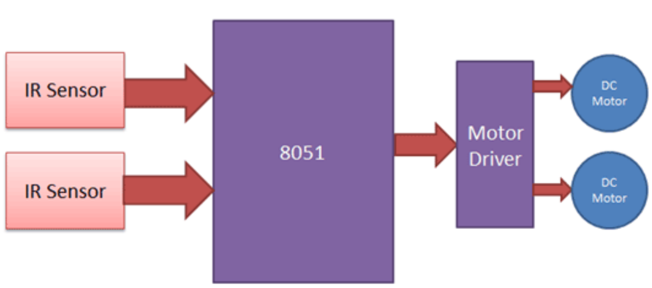
Concept of line follower is related to light. We have used the behavior of light at black and white surface. When light fall on a white surface it will almost fully reflects and in case of black surface light is absorbed by black surface. This explained behavior of light is used in this line follower robot.



In this line follower robot project, we have used IR Transmitters and IR receivers also called photo diodes for sending and receiving light. IR transmits infrared lights. When infrared rays fall on white surface, it is reflected and recevied by photodiode and generates some voltage changes. When IR light falls on black surface light is absorbed by the black surface and not rays reflect, so photo diode did not receive any light or rays. Here in this line follower robot when sensor senses white surface then microcontroller gets 0 as input and when senses black line microcontroller gets 1 as input.

• Circuit explanation:

We can divide the whole line follower robot into various sections like sensor section, control section and driver section.



Sensor section:

In a typical line follower robot setup with an IR sensor detecting a black line on a white surface:

1. ***Output of 1 (or HIGH): The sensor outputs a 1 (or HIGH) when it is directly over the black line. This is because the black line absorbs more infrared light, and the sensor detects less reflected light, signaling the presence of the line.***

2. ***Output of 0 (or LOW): The sensor outputs a 0 (or LOW) when it is not over the black line. In this case, the sensor is likely detecting more reflected infrared light from the white surface, indicating that the robot is away from the black line.***

Control Section:

8051 microcontroller is used for controlling whole the process of line follower robot. The outputs of sensors are connected to pin number P1.0 and P1.1 of 8051. 8051 reads these signals and send commands to driver circuit to drive line follower.

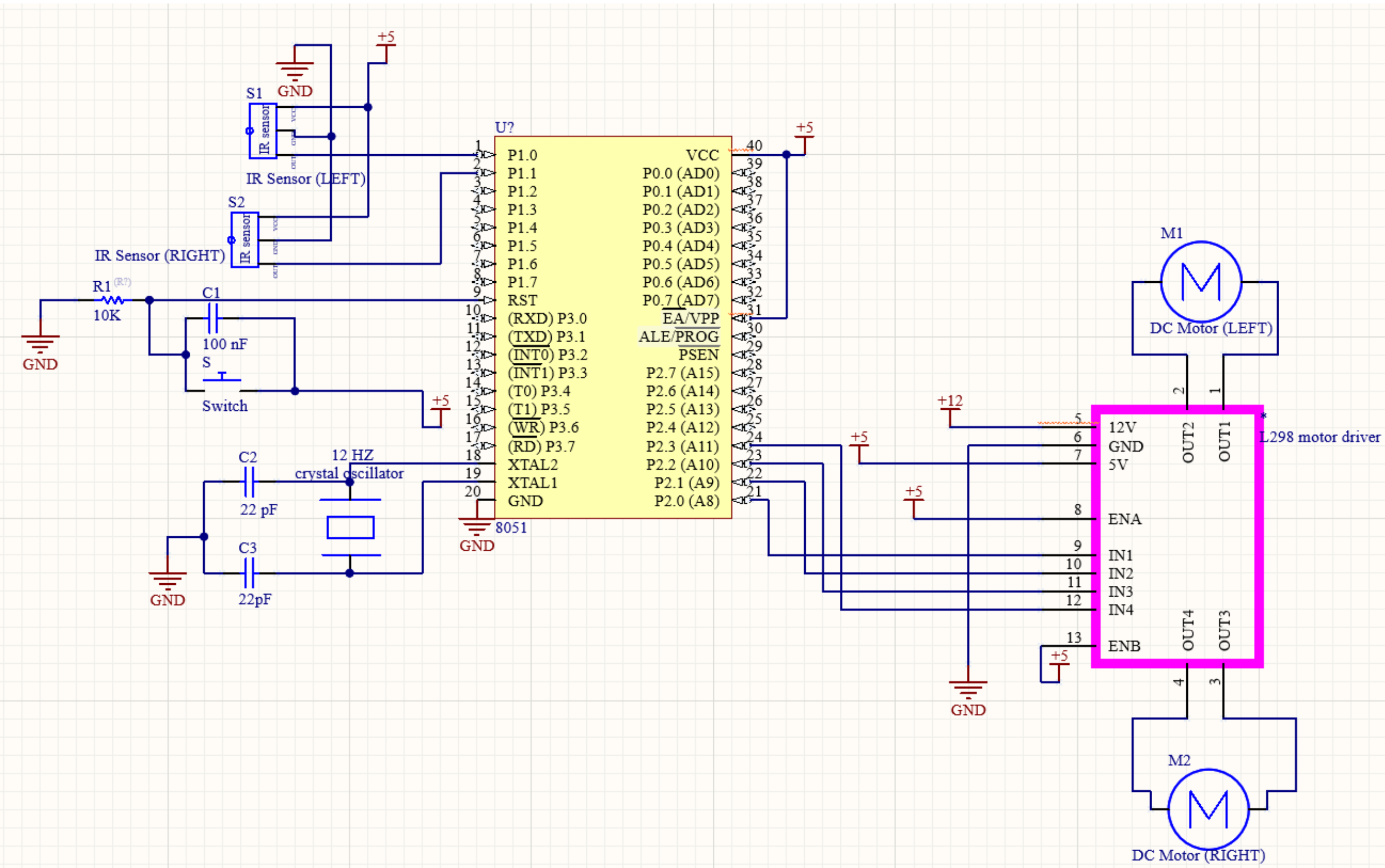
The line follower robot's control system interprets these sensor outputs and uses them to make decisions about the robot's movement. For example, when a sensor detects the black line (output of 1), the control system might instruct the robot to adjust its direction to stay on the line. When the sensor does not detect the black line (output of 0), the control system might take corrective action to bring the robot back onto the line.

Driver section:

Driver section consists of motor driver and two DC motors. Motor driver is used for driving motors because microcontroller does not supply enough voltage and current to motor. So, we added a motor driver circuit to get enough voltage and current for motor. Microcontroller sends commands to this motor driver and then it drive motors.

• 8051 Based Line Follower Robot Circuit:

Circuit is very simple for this line follower robot. Output of sensors is directly connected to pin number P1.0 and P1.1 of microcontroller. And motor driver's input pin 1 , 2, 3 and 4 is connected at pin number P2.1, P2.2, P3.1 and P2.4 respectively. And one motor is connected at output pin of motor driver 1 and 2and another motor is connected at 3 and 4.



• The algorithm [logic] of line follower robot:

Left Sensor	Right Sensor	Left Motor		Right Motor		Action
P1.0	P1.1	P2.0	P2.1	P2.2	P2.3	
0	0	0	1	0	1	Move Forward
0	1	0	1	0	0	Move Right
1	0	0	0	0	1	Move Left
1	1	0	0	0	0	Stop

● Code:

```
ORG 0000H


   SETB P1.0 ;config p1.0 as input
   SETB P1.1 ;config p1.1 as input


   MOV P2,#00H ;config p2 as output


MAIN:
;checking for sensors readings to perform the required movement
   JNB P1.0, RIGHT
   JNB P1.1, LEFT
   MOV p2, #00H
   SJMP MAIN



LEFT:
;moving left to move the right sensor away from the line
   CLR P2.0
   CLR P2.1
   CLR P2.2
   SETB P2.3
   SJMP MAIN
```

```
RIGHT:
;moving right to move the left sensor away from the line
   JNB P1.1, FORWARD
   CLR P2.0
   SETB P2.1
   CLR P2.2
   CLR P2.3
   SJMP MAIN

FORWARD:
   CLR P2.0
   SETB P2.1
   CLR P2.2
   SETB P2.3
   SJMP MAIN

RET
END
```

• Snippets and explanation:

1. Input / Output configuration:

```
SETB P1.0 ;config p1.0 as input
SETB P1.1 ;config p1.1 as input

MOV P2,#00H ;config p2 as output
```

The pins of P1 have been selected to be connected to the 2 IR sensors so both pins P1.0 and P1.1, which the sensor been attached to (P1.0 for the left sensor and P1.1 for the right sensor) initialized and configured as input.

The pins of P2 have been selected to be connected to the motor driver pins so it has been initialized as output.

2. Main loop and logic operation:

```
MAIN:
;checking for sensors readings to
    JNB P1.0,RIGHT
    JNB P1.1,LEFT
    MOV p2, #00H
    SJMP MAIN
```

The actual start of the program has been labeled as MAIN, the status of both sensor pins is being checked as first the status of the P1.0 (left sensor is being checked), if the pin is low that means the left sensor did not interfere with the line, so we assume that the right sensor did, and program performs a jump to the label RIGHT.

3. The right sensor checks in the Right function:

```
RIGHT:
;moving right to move the left sen
    JNB P1.1,FORWARD
    CLR P2.0
    SETB P2.1
    CLR P2.2
    CLR P2.3
    SJMP MAIN
```

After checking the left sensor using JNB and finds that P1.0 did not read the line so the state is zero, the program performs the jump to label RIGHT as it assumed that if the left sensor didn't read the line so the right one did, now the program needs to check if the assumption is correct or not before performing the right movement, as shown in the snippet up there the right sensor is checked using the JNB at the pin P1.1 as well and if it finds the pin at zero state it means that neither the left nor the right sensor read the black line so it jumps to the label FORWARD to perform the normal forward movement.

4. Forward movement:

```
FORWARD:
    CLR P2.0
    SETB P2.1
    CLR P2.2
    SETB P2.3
    SJMP MAIN
```

when the program finds out that both pins attached to the sensors are at LOW state which means no line been detected by both the program sets pins P2.1 and P2.3 to give signals to the motor driver to move forward and after performing the action the SJMP moves the program back to the MAIN label to perform the check of the sensors again.

5. RIGHT label and right movement:

```
RIGHT:
;moving right to move the left sen
    JNB P1.1,FORWARD
    CLR P2.0
    SETB P2.1
    CLR P2.2
    CLR P2.3
    SJMP MAIN
```

Getting back to the RIGHT label, assuming that the JNB did not jump the code to the FORWARD label as the state of P1.1 is HIGH which means the right sensor did find the black line, now the robot needs to modify its movement action to get the right sensor away from the line so the P2.1 pin sends a HIGH signal to the driver while the other three pins sends a LOW signal which stops the right motor and lets the left motor to move, this will move away the right sensor from the line repositioning the robot, then SJMP MAIN restarts the process from the MAIN label again.

6. MAIN label assuming first jump not performed:

```
MAIN:
;checking for sensors readings to
    JNB P1.Ø,RIGHT
    JNB P1.1,LEFT
    MOV p2, #ØØH
    SJMP MAIN
```

Assuming the JNB at the first line is not performed which means the HIGH state of P1.0 (left sensor read the line) so the program moves to the next line which is checking the state of the P1.1 as if it gives HIGH as well that means both lines are on the line which means the robot reached to a black area or spot on the ground so it stops immediately by performing a LOW state on all the pins of P2 sending LOW signals to the motor driver to stop the movement of the robot and the SJMP MAIN restarts the process from the MAIN label.

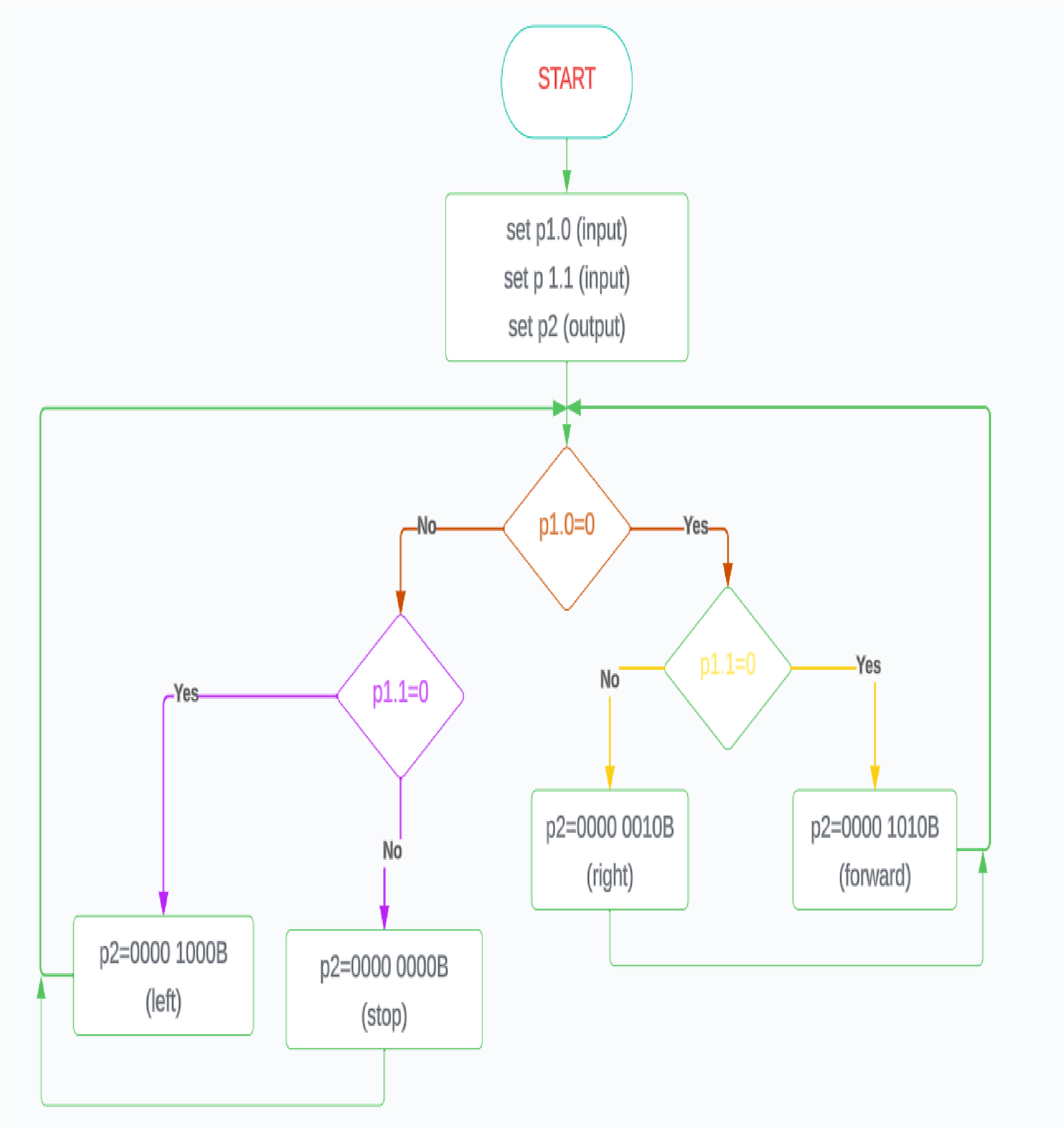
But if the second JNB here happened which means the right sensor did not meet the line so the state of P1.1 pin is LOW and the state of P1.0 which had been already checked is HIGH that means the left sensor is on the line, so the program jumps to the LEFT label.

7. LEFT label and left movement:

```
LEFT:
;moving left to move the right sen
    CLR P2.Ø
    CLR P2.1
    CLR P2.2
    SETB P2.3
    SJMP MAIN
```

After finding out that the Left sensor is on the line the robot needs to modify itself to move the left sensor away from the line, the P2.3 is being set to HIGH to send a HIGH signal to the driver to perform a forward movement to the right motor while the other pins of P2 is being cleared to set a LOW signal to the other three input pins of the driver stopping the left motor, then SJMP MAIN restarts the process from the MAIN label.

FLOW CHART



START
set p1.0 (input)
set p 1.1 (input)
set p2 (output)
p1.0=0
No
Yes
p1.1=0
Yes
No
p1.1=0
No
Yes
p2=0000 1000B
(left)
p2=0000 0000B
(stop)
p2=0000 0010B
(right)
p2=0000 1010B
(forward)

• Final product: