“Obstacle avoiding car”

Name	Section
Taha Ibrahim Taha	4
Gamal Adel Gamal	3
Soliman Mahmoud Soliman	4
Abdel haleem Mostafa Mohamed	4
Abdelrahaman Mohamed Abdelhamed	5
Alaa Ahmed Mohamed	2
Asmaa Reda Mohamed	2
Elsayed Abbas Elsayed	2

Supervised by: Dr/Ahmed shalaby, Eng / Ghada Refaaye

Introduction:

Obstacle avoidance is one of the most important aspects of mobile robotics. Without it, robot movement would be very restrictive and fragile. This project proposes a robotic vehicle with intelligence built into it to direct itself whenever an obstacle comes in its path. So, to protect the robot from any physical damage. This can be designed to build an obstacle avoidance robotic vehicle using ultrasonic sensors for its movement.

The primary objectives:

1. Sensing and Perception:

Implementing sensors such as ultrasonic to detect obstacles and the surrounding environment.

2. Navigation and Path Planning:

Designing algorithms for determining a safe and obstacle-free path for the robot to follow.

3. Control System:

Developing a control system that enables the robot to act on the information gathered from sensors and execute appropriate movements.

4. Autonomous Decision-Making:

Designing decision-making algorithms to allow the robot to make real-time decisions about its movements, such as stopping, changing direction, or maneuvering around obstacles.

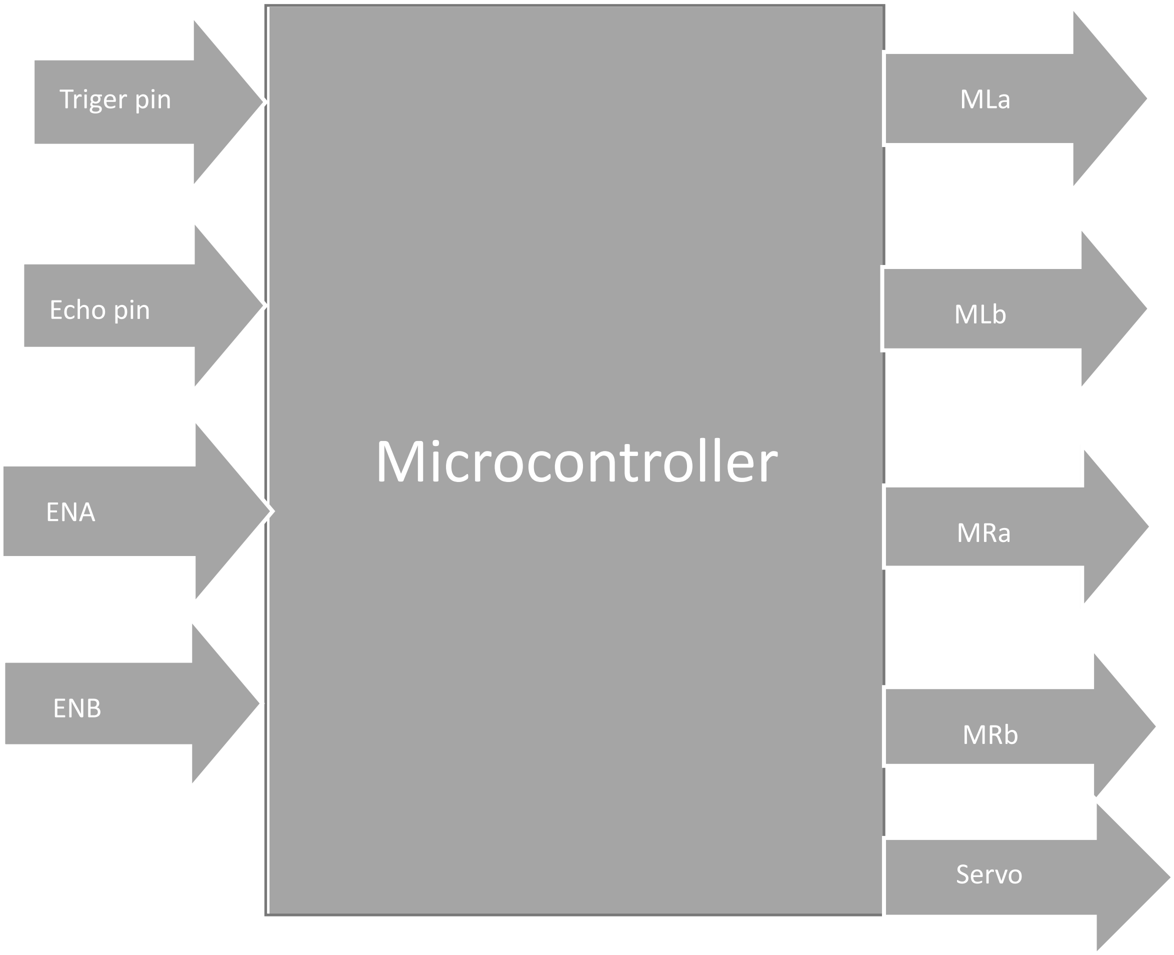
5. Energy Efficiency:

Optimizing the robot's movements and decision-making processes to conserve energy and extend operational time.

Goals:

The goal of an obstacle-avoiding robot project is to create a reliable and efficient robotic system capable of autonomously navigating its environment while intelligently avoiding obstacles in its path.

Block Diagram:



Inputs/Outputs:

Input name	Pin number	meaning
trigger Pin	9	This is the trigger pin of the ultrasonic sensor. It sends out a pulse to measure the distance to an obstacle.
echo Pin	8	This is the echo pin of the ultrasonic sensor. It receives the reflected pulse from the obstacle and calculates the distance.
Servo Motor	10	The readings from the ultrasonic sensor (distance measurements) are used as inputs to make decisions about the robot's movement.
ENA	2	Indicate the speed of the motors using pulse-width modulation (PWM), controls the speed of the left motor.
ENB	3	Indicate the speed of the motors using pulse-width modulation (PWM), controls the speed of the right motor.

Output name	Pin number	meaning
MLa	4	This is the first pin of the left motor. It controls the direction of the motor.
MLb	5	This is the second pin of the left motor. It controls the direction of the motor.
MRa	6	This is the first pin of the right motor. It controls the direction of the motor.
MRb	7	This is the second pin of the right motor. It controls the direction of the motor.
Servo	10	The servo motor is primarily used to control the rotation of the ultrasonic sensor, allowing the sensor to look in different directions.

Hardware Components:

Arduino Uno	The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.
Motor Driver	It provides the power needed to control motors and other components within an application, The motor driver is essentially responsible for providing the voltage, current, directionally and protection necessary to efficiently operate your robotic parts or devices
Battery 12v	A 12-volt battery gets your engine going and helps keep it running. Its main role is to start the car. When you push the ignition button or turn the ignition key
4 wheels and it's 4 motors	They take a power from motor driver then move the wheels
Servo Motor 180 Degree	is an electrical device which can push or rotate an object with great precision and it is kind of actuator, A servo motor can turn 180 degrees
Ultrasonic sensor with its holder battery 9v	As the name indicates, Ultrasonic sensors measure distance by using ultrasonic waves and it is composed of Receiver and transmitter and the width of received pulse is used to calculate the distance from the reflected object and the sensor measures the time taken by the signal to travel between the transmission of the sound by the transmitter to the reflecting back towards the receiver. Ex: HC-SR04
Jumper male to male	provides an easy way for you to build your own circuit on the board as it is used to interconnect the components of a breadboard or other prototype or test circuit internally or with other equipment or components without soldering and it's ends have a pin protruding and can plug into things
Jumber male to female	provides an easy way for you to build your own circuit on the board as it is used to interconnect the components of a breadboard or other prototype or test circuit internally or with other equipment or components without soldering and its ends don't and are used to plug things into
Arduino Uno cable	Use it to connect Arduino Uno, Arduino Mega 2560, Arduino 101 or any board with the USB female A port of your computer and transform data to and from Arduino
Arduino ide	Enables you to write and edit code and convert this code into instructions that Arduino hardware understands
Atmel Studio	Atmel Studio is an integrated development platform for Atmel AVR and ARM microcontrollers as we write in it with c language and then transform to machine language then send to AVR

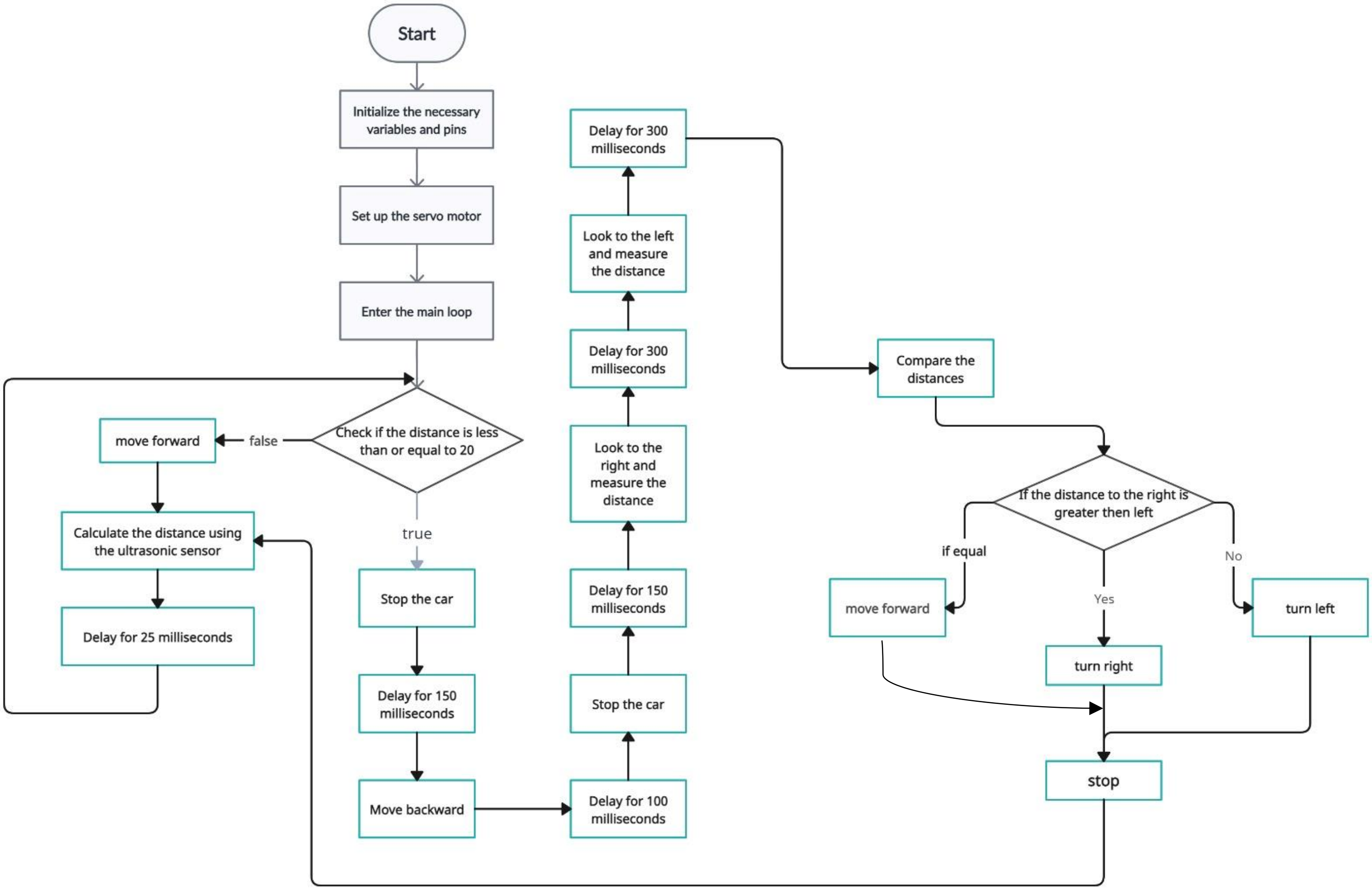
Microcontroller	s an integrated circuit consisting of a complete computer on a single chip and used for specified control functions
Sensor	Is a device that detects some type of input from the physical environment? ex: DHT11, PIR

Algorithm

First, we Check Distance: If the current distance is less than or equal to 15 units:

```
    1 – Stop the Car (for 150 ms)
    2 – Move Backward (for 100 ms)
    3 – Stop the Car again (for 150 ms)
    4 - Measure Distances to the Right and Left:
        1 - If distanceRight is greater than distanceLeft:
            1 - Turn the car to the right. (for 225 ms)
            2- Stop the Car
        2 - Else, if distanceRight is less than distanceLeft:
            1 - Turn the car to the left. (for 250 ms)
            2 – Stop the Car
        3 - Else (if both distances are equal):
            Move the car forward.
    5 – Move forward: If the initial distance check was false (greater than 15 units), move the car forward
    6 - Recalculate Distance
    7 – Delay (for 5 ms), this is to avoid Hardware Malfunctions and to give enough time for car to start checking again
```

Flow Chart:


Start
Initialize the necessary variables and pins
Set up the servo motor
Enter the main loop
Check if the distance is less than or equal to 20
false
move forward
Calculate the distance using the ultrasonic sensor
Delay for 25 milliseconds
true
Stop the car
Delay for 150 milliseconds
Move backward
Delay for 100 milliseconds
Stop the car
Delay for 150 milliseconds
Look to the right and measure the distance
Delay for 300 milliseconds
Look to the left and measure the distance
Delay for 300 milliseconds
Compare the distances
If the distance to the right is greater then left
if equal
move forward
Yes
turn right
No
turn left
stop

Ultrasonic code in C language:

```c
#define F_CPU 16000000

#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdio.h>

#define TRIG_PIN 3
#define ECHO_PIN 2

long t = 0;
uint32_t ms_counter = 0;


int main() {
        setup();
        pulse_in_init();

        while (1) {
                loop();
        }

        return 0;
}

void setup() {
        //output
        DDRD |= (1 << TRIG_PIN);

        //input
        DDRD &= ~(1 << ECHO_PIN);

        //Rising edge
        EICRA |= (1 << ISC00) | (1 << ISC01);

        //External Interrupt 0
        EIMSK |= (1 << INT0);

        //Set enable
        sei();
}


void loop() {

        PORTD &= ~(1 << TRIG_PIN);
        _delay_us(2);
        PORTD |= (1 << TRIG_PIN);
        _delay_us(10);
        PORTD &= ~(1 << TRIG_PIN);

        _delay_ms(50);

        t = pulse_in();
        int distance = (int)(0.5 * t * 0.0343);

        printf("Distance: %d cm\n", distance);
        _delay_ms(500);
}

ISR(INT0_vect) {
        static long startTime = 0;

        if (PIND & (1 << ECHO_PIN)) {
                startTime = pulse_in();
                } else {
                t = pulse_in() - startTime;
        }
}


void pulse_in_init() {

        //Normal mode operation
        TCCR0A = 0x00;

        //No prescaling
        TCCR0B |= (1 << CS00);

        //Timer0
        TIMSK0 |= (1 << TOIE0);

        //Set enable
        sei();
}

uint32_t pulse_in() {
        uint32_t m;
        uint8_t sreg = SREG;

        // Disable interrupts
        cli();
        m = ms_counter + TCNT0;
        SREG = sreg;

        return m;
}


ISR(TIMER0_OVF_vect) {
        //Overflow handler
        ms_counter += 256;
}
```

Test Strategy

First, we change the Parameters, arguments and the environment in which our car interacts in a random way and then we notice what change occurs in the real world, as we change the angle of the Ultrasonic sensor , change the delays of each function and also modify the algorithm to get the best algorithm (with the best parameters) that makes the car avoid the obstacles fast , we will We will show you some of the Test cases that we have tried to ensure that the car works correctly in different environments and types of obstacles:

First obstacle position	Distance before obstacle (Cm)	Distance between the two obstacles (Cm)	Successfully Passed	Number of Seconds
Right	20	50	Yes	26
Right	25	40	Yes	29
Right	30	30	Yes	20
Left	20	50	Yes	29
Left	25	40	Yes	22
Left	30	30	Yes	23

* The number of seconds depends on the strength of the battery charge and the floor on which the car is moving, whether it is ceramic, wood, or anything else.