Obstacle Avoider Wheeled Robot Using STM32 Microcontroller and Ultrasonic Sensors

Contributors:

Fatma Alagroudy 21-101154
Kerollos Ehab 21-101137
Manuel Ashraf 21-101148
Yara Abbas 21-101151
Youssef Nagah 21-101139

Abstract: This report details the development of an obstacle-avoiding wheeled robot using an ARM STM32 microcontroller. The project aims to create a robot capable of autonomous navigation by detecting and avoiding obstacles using ultrasonic sensors. Key components include the STM32 microcontroller, HC-SR04 ultrasonic sensors, L298N motor drivers, a LCD display, and a buzzer. The robot's design, hardware and software requirements, working principle, and the implemented algorithm are discussed in detail. This project demonstrates the integration of various hardware components with embedded software to achieve efficient obstacle avoidance in real-time.

1. **Introduction**
   In our increasingly automated world, intelligent machines that can navigate their surroundings safely and efficiently are becoming ever more crucial. This report details the development of such a machine: an obstacle-avoiding car. This car utilizes an STM32 microcontroller as its brain, HC-SR04 ultrasonic sensors as its eyes, and an L298N motor driver as its muscles. Through this combination of hardware, the car can detect obstacles in its path and autonomously adjust its course to avoid them. This report will delve into the design choices, the hardware and software components used, the working principle behind the obstacle

avoidance, and the algorithm that governs the car's decision-making process.

2. Hardware and Software Requirements

This project was developed using the following hardware and software components to ensure its functionality and performance.

Hardware:

- STM32 Microcontroller
- 3 Ultrasonic Sensors (HC-SR04)
- Motor Drivers (L298N)
- DC Motors
- LCD Display
- Buzzer
- Power Supply (Battery pack)
- Connecting wires and breadboard
- 4 Wheels
- Chassis

Software:

- STM32CubeIDE
- STMCubeProgrammer

3. Design and Modeling of Robot Vehicle

This section will examine the robot vehicle's design and modelling for avoiding obstacles. This includes a thorough examination of the essential parts that make up the robot's design and operation.These parts are: STM32 microcontroller, ultrasonic sensors, motor drivers, LCD display, and buzzer. Each part is essential to the robot's ability to navigate on its own by recognising and avoiding obstacles instantly.

3.1. STM32 Microcontroller

The STM32 microcontroller serves as the central processing unit of the robot. It integrates various peripherals required for the project, including GPIO ports for sensor and motor connections and timers. These are essential for interfacing with sensors, motors, and other components. The specific model used in this project is STM32F103C8T6. Figure 1 represents a visual for the STM32

Microcontroller we used.

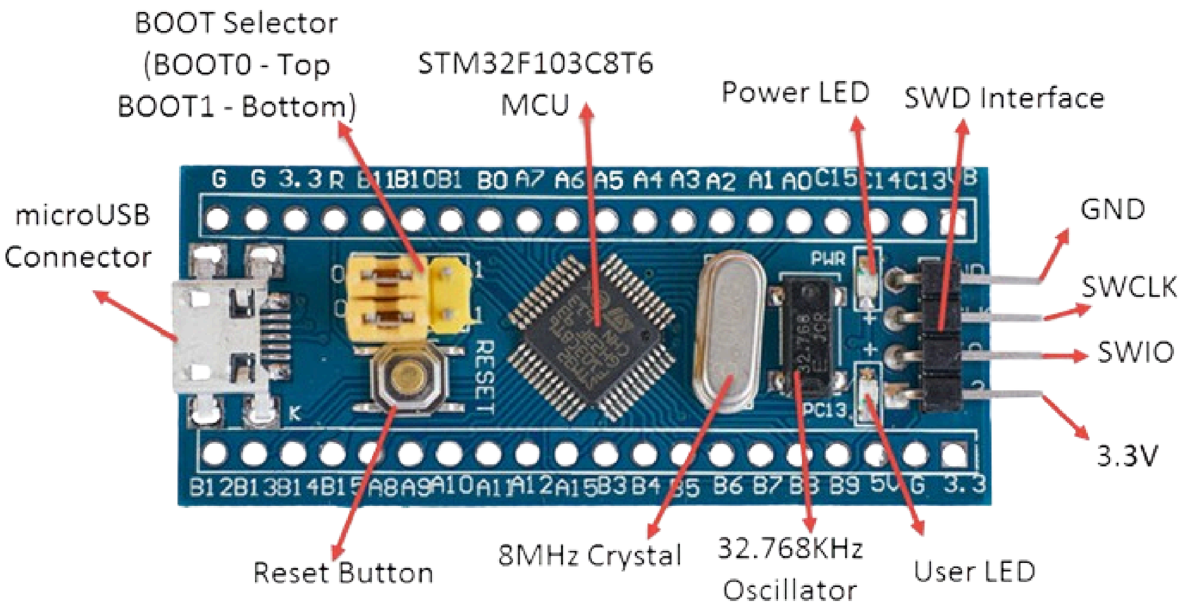

Figure 1. STM32 (Source: Pelayo)

3.3. Ultrasonic Sensors

Ultrasonic sensors (HC-SR04) are used to detect obstacles in the robot's path. These sensors work by emitting ultrasonic waves and measuring the time it takes for the echo to return after bouncing off an obstacle.

Figure 2. Ultrasonic sensor (HC-SR04) (Source: “HC SR04 Datasheet and Pinout – Ultrasonic Sensor Noncontact Range Detection”)

3.4. Motor Drivers

The L298N motor driver module controls the speed and direction of the DC geared motors. It allows for independent control of two motors, enabling differential steering for the robot.

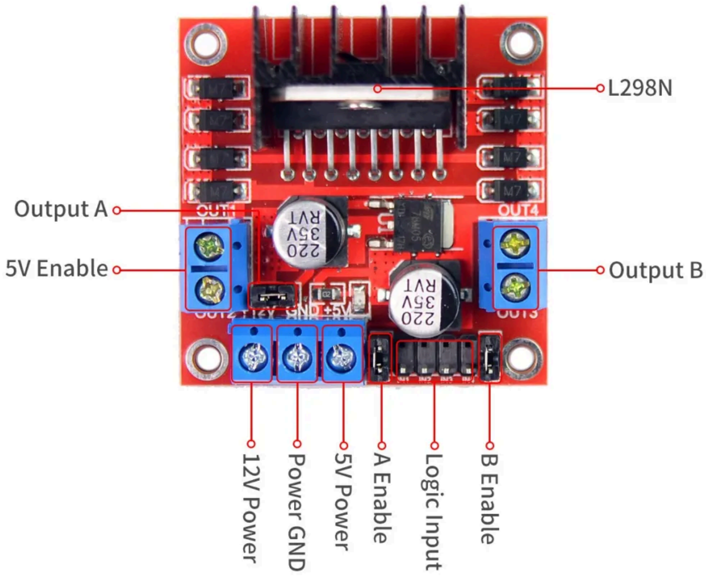

Figure 3. Motor Driver (L298N) (Source: "L298N Datasheet and Pinout")

3.5. LCD

The LCD provides the reading distance for the front ultrasonic sensor .

3.6. Buzzer

The buzzer serves as an alarm .When the robot is still searching for a free pathway to walk through, the buzzer beeps.

3.7. Battery

A voltage supply of 9V that powers the entire circuit and motors.

4. Algorithm and Working Principle

The steps we took to build this project were as follows:

1)Assemble the hardware : We figured out the required components and wired them into the robot's body .

2)Write the code: We wrote the algorithm and the code required for the robot to detect obstacles and find a free path to walk through.

The Algorithm we used was:
When the robot is powered on using a switch , it starts moving forward until the front sensor detects the distance ahead is less than 50 cm.When it approaches an obstacle it stops for 300ms and the right and left sensors start reading. The robot then compares right distance to the left distance and it walks in the direction where the distance is greater .After that, the algorithm keeps repeating itself so that the robot is always moving , measuring distances around and finding the clearest path.

3)Upload the code to the microcontroller: Here is a copy of the main code .

```
move_forward();
    DistanceF = read_front();
    while(DistanceF < 50)
    {
      HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_SET); //Buzzer
      stop();
      DistanceR = read_right();
      DistanceL = read_left();
      if (DistanceR > DistanceL)
      {
        turn_right();
        stop();
      }
      else
      {
      turn_left();
      stop();
      }
      DistanceF = read_front();
    }
```

The Functions we used:

```
int read_right(void)
{
  HAL_GPIO_WritePin(TRIG_PORT_R, TRIG_PIN_R, GPIO_PIN_SET);  // pull
the TRIG pin HIGH
  __HAL_TIM_SET_COUNTER(&htim3, 0);
  while (__HAL_TIM_GET_COUNTER (&htim3) < 10);  // wait for 10 us
  HAL_GPIO_WritePin(TRIG_PORT_R, TRIG_PIN_R, GPIO_PIN_RESET);  // pull
the TRIG pin low

  pMillis_R = HAL_GetTick(); // used this to avoid infinite while loop
(for timeout)
  // wait for the echo pin to go high
```

```
  while (!(HAL_GPIO_ReadPin (ECHO_PORT_R, ECHO_PIN_R)) && pMillis_R +
10 >  HAL_GetTick());
  Value1R = __HAL_TIM_GET_COUNTER (&htim3);

  pMillis_R = HAL_GetTick(); // used this to avoid infinite while loop
(for timeout)
  // wait for the echo pin to go low
  while ((HAL_GPIO_ReadPin (ECHO_PORT_R, ECHO_PIN_R)) && pMillis_R + 50
> HAL_GetTick());
  Value2R = __HAL_TIM_GET_COUNTER (&htim3);

  DistanceR = (Value2R-Value1R)* 0.034/2;
  return DistanceR;
}
```

```
void move_forward(void)
{
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_7, GPIO_PIN_SET);
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_5, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_12, GPIO_PIN_SET);
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_11, GPIO_PIN_RESET);
}
void turn_right(void)
{
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_5, GPIO_PIN_SET);
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_7, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_12, GPIO_PIN_SET);
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_11, GPIO_PIN_RESET);
  HAL_Delay(300);
}
void turn_left(void)
{
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_7, GPIO_PIN_SET);
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_5, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_11, GPIO_PIN_SET);
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_12, GPIO_PIN_RESET);
  HAL_Delay(300);
}

void stop(void)
{
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_7, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_5, GPIO_PIN_RESET);
```

```
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_11, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_12, GPIO_PIN_RESET);
}
```

4)Test and debug: To ensure the robot was working effectively , we needed to keep testing and debugging until we reached the final working model / prototype . Figure 3 represents our working prototype

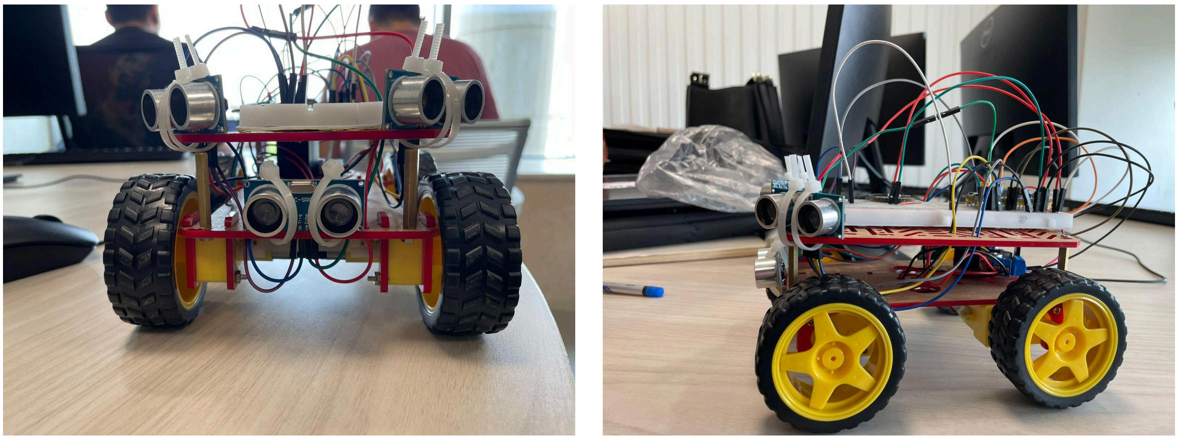

Figure 4. Working Prototype

5. Conclusion

In conclusion, the development of the obstacle-avoiding wheeled robot using the STM32 microcontroller and ultrasonic sensors has demonstrated the practical application of embedded systems in autonomous navigation. By integrating the STM32F103C8T6 microcontroller, HC-SR04 ultrasonic sensors, L298N motor drivers, an LCD display, and a buzzer, the robot effectively detects and avoids obstacles in its path. The project highlighted the importance of precise sensor data processing, real-time decision-making, and robust motor control to achieve efficient obstacle avoidance. This autonomous system has potential applications in various fields, such as warehouse automation, search and rescue missions, and home automation. The successful implementation of this project underscores the capabilities of the STM32 microcontroller in managing complex tasks and achieving reliable performance in real-time applications.

References

- “HC SR04 Datasheet and Pinout – Ultrasonic Sensor Noncontact Range Detection.” *NetSonic*, 25 June 2021, https://netsonic.fi/en/hc-sr04-datasheet-and-pinout-ultrasonic-sensor-noncontact-range-detection/
- Pelayo, Roland. “Getting Started with Blue Pill and STM32Cube.” *Microcontroller Tutorials*, 2 October 2020, https://www.teachmemicro.com/getting-started-blue-pill-stm32cube/
- “L298N - Motor.| Micropeta ” Www.micropeta.com, www.micropeta.com/video20
- MicroPeta by Nizar Mohideen. “42. STM32CubeIDE Ultrasonic Distance Sensor. HC-SR04 with STM32F103C8T6.” YouTube, 23 Aug. 2021, www.youtube.com/watch?v=oD0mc6FXQ0s&t=31s
- Goswami, Shankha Shubhra, and Sushil Kumar Sahoo. “Design of a Robotic Vehicle to Avoid Obstacle Using Arduino Microcontroller and Ultrasonic Sensor.” Advances in transdisciplinary engineering, 2023.