Title:

Building an Obstacle Detection System.

Theory & Methodology:

When embarking on the development of electronic products, selecting the appropriate microcontroller is paramount. Several factors come into play, including the availability of essential tools such as assemblers, debuggers, efficient C language compilers, emulators, technical support, and access to expertise both internally and externally. In light of these considerations, the Arduino series emerges as a favorable choice due to its extensive online community support and the abundance of freely available and efficient code samples from diverse projects.

Arduino, an open-source platform, offers a seamless environment for creating interactive electronics projects. It comprises a programmable microcontroller and an Integrated Development Environment (IDE) that facilitates writing and uploading computer code to the microcontroller board. A notable advantage of Arduino Uno, one of the popular boards in the Arduino series, is its simplicity and the absence of a requirement for additional hardware such as a programmer or burner to upload new code. Utilizing a USB cable and the Arduino IDE, which employs a simplified version of the C++ programming language, users can effortlessly load code into the board.

Coding a Simple Obstacle Detection System in Arduino IDE phase entails writing code within the Arduino IDE to develop the logic for an obstacle detection system. The code will interface with sensors and peripherals, manage data processing, and control output devices based on predefined conditions.Once the code is crafted, it is uploaded to the Arduino Uno board via a USB cable. The hardware setup involves connecting the Arduino board to sensors, such as the sonar sensor (HCSR04), and an LED to indicate the detection of obstacles. Pin configurations and hardware connections are established to enable the system to function effectively.To validate the functionality of the code and hardware configuration, simulation tools like Proteus and Tinkercad are utilized. These platforms facilitate virtual testing of the system, allowing for observation of how the system behaves in response to different inputs and scenarios.

In summary, this laboratory experiment aims to provide participants with hands-on experience in developing an obstacle detection system using Arduino microcontrollers. Through the practical application of coding, hardware implementation, and simulation, participants gain insight into microcontroller-based systems and their real-world applications.

Overview Of Arduino Uno Board:

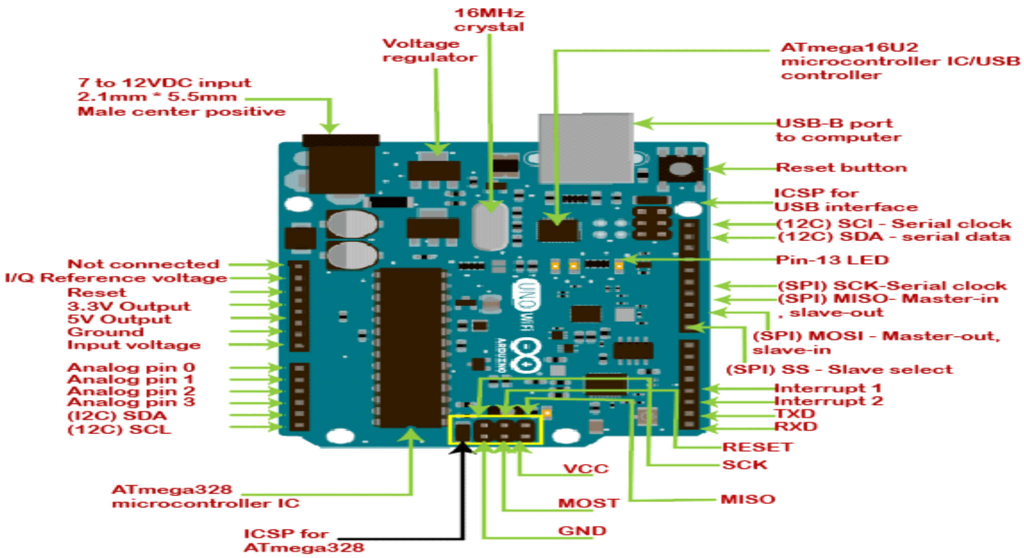

Figure: Pinout diagram of an Arduino Uno Board

Overview Of Sonar Sensor:

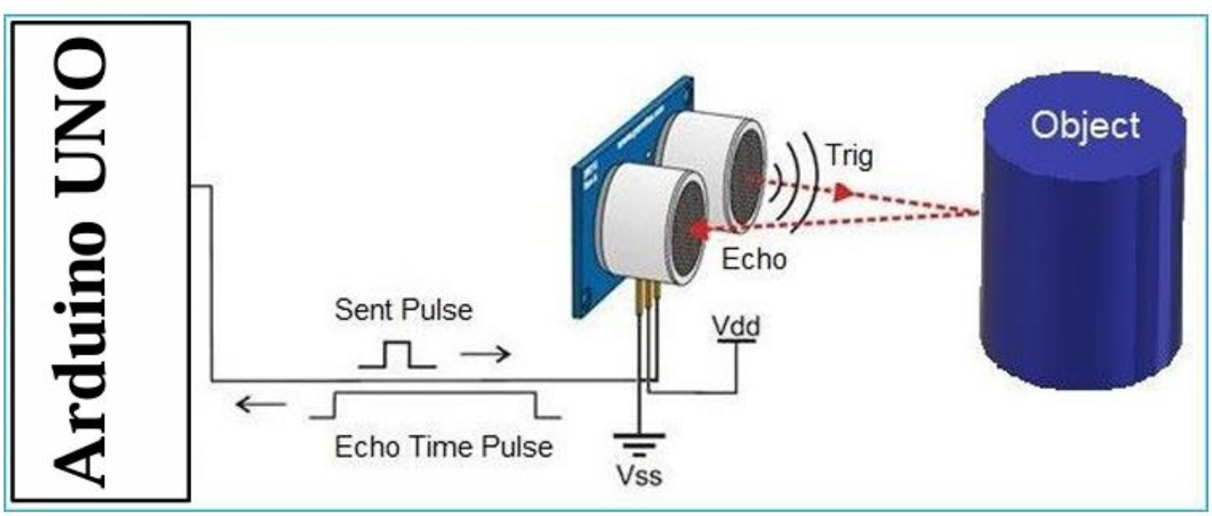

Figure: Diagram of a Sonar Sensor

Apparatus:

- Arduino IDE (any version)
- Arduino Uno (R3)
- Breadboard
- Sonar Sensor (HCSR04)
- LED
- Resistor(s)
- Connecting Wire

Experimental Procedure:

First, we understood the theory and methodology of an Arduino Uno board. Then we collected an Arduino Uno with some LED's, resistors, sonar sensor and connecting wires. After that, the Arduino Uno R3 board was connected to the computer via USB cable. Subsequently, a blank sketch was opened in the Arduino IDE, and the provided code for the obstacle detection system was written in the sketch. The code included initializing pins for the ultrasonic sensor, trigger, and LED, as well as defining functions to convert microseconds to inches and centimeters.

Following the code writing, the sketch was saved with an appropriate file name. The code was then compiled and verified for errors using the Verify/Compile option in the Arduino IDE. Once compiled successfully, the code was uploaded to the Arduino Uno board by selecting the correct board type and port in the Arduino IDE and clicking on the upload option.

After uploading the code, the familiarity with Arduino commands was ensured, which included commands such as pinMode(), digitalWrite(), delayMicroseconds(), and Serial.print(). These commands were essential for configuring pins, providing input/output signals, introducing delays, and printing text to the serial monitor.

Next, the circuit setup for the obstacle detection system was described. The main components involved were the Arduino Uno board, the ultrasonic sensor (HCSR04), and an LED. The ultrasonic sensor was connected to digital pins on the Arduino for trigger and echo, while the LED was connected to indicate obstacle detection.

Once the hardware setup was complete, the simulation of the code in Thinkercad was explained. This involved adding the Arduino and Ultrasonic Sensor libraries to Proteus, setting up the circuit simulation, and running the simulation to observe the behavior of the obstacle detection system.

Additionally, instructions for simulating the code in Tinkercad were provided. Tinkercad, a free online 3D design program, allowed for simulating the Arduino circuit with the ultrasonic sensor and LED. The steps included building the LED circuit, adding the proximity sensor, and running the simulation to observe the distance measurement and LED behavior.

<u>Hardware Picture(s):</u>

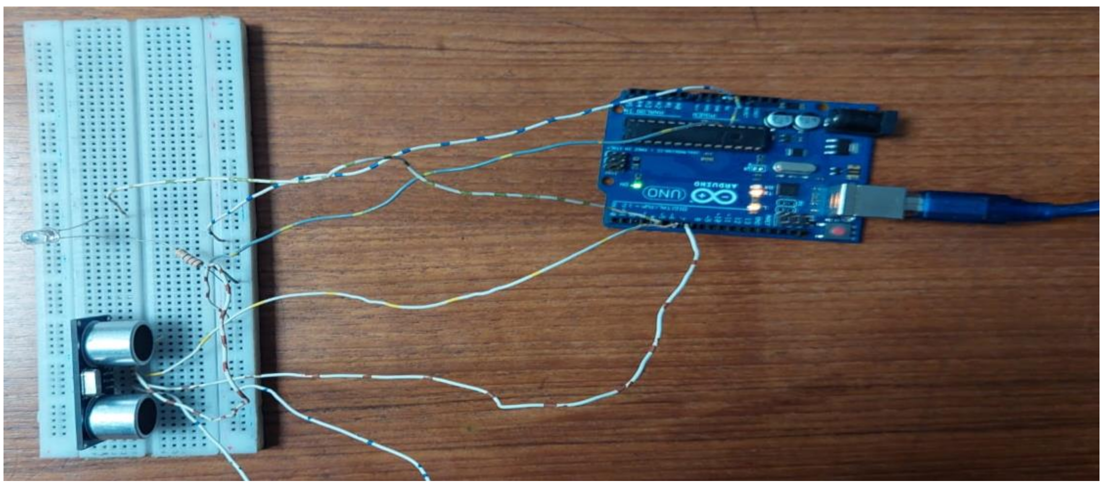

Figure: Without obstacles LED is OFF

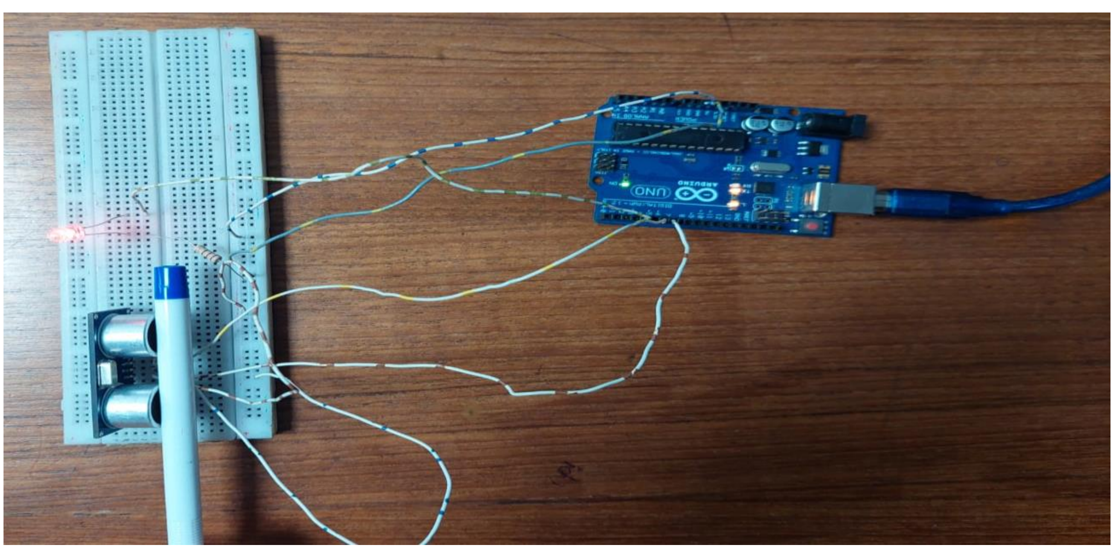

Figure: With obstacles LED is ON

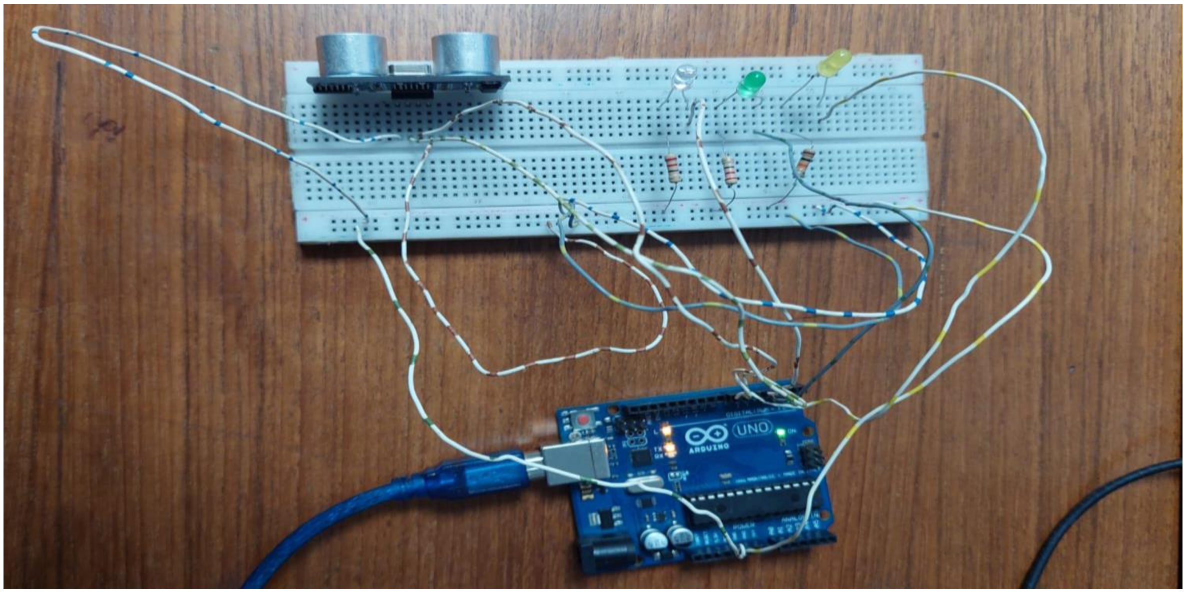

Figure: Without obstacles LED is OFF

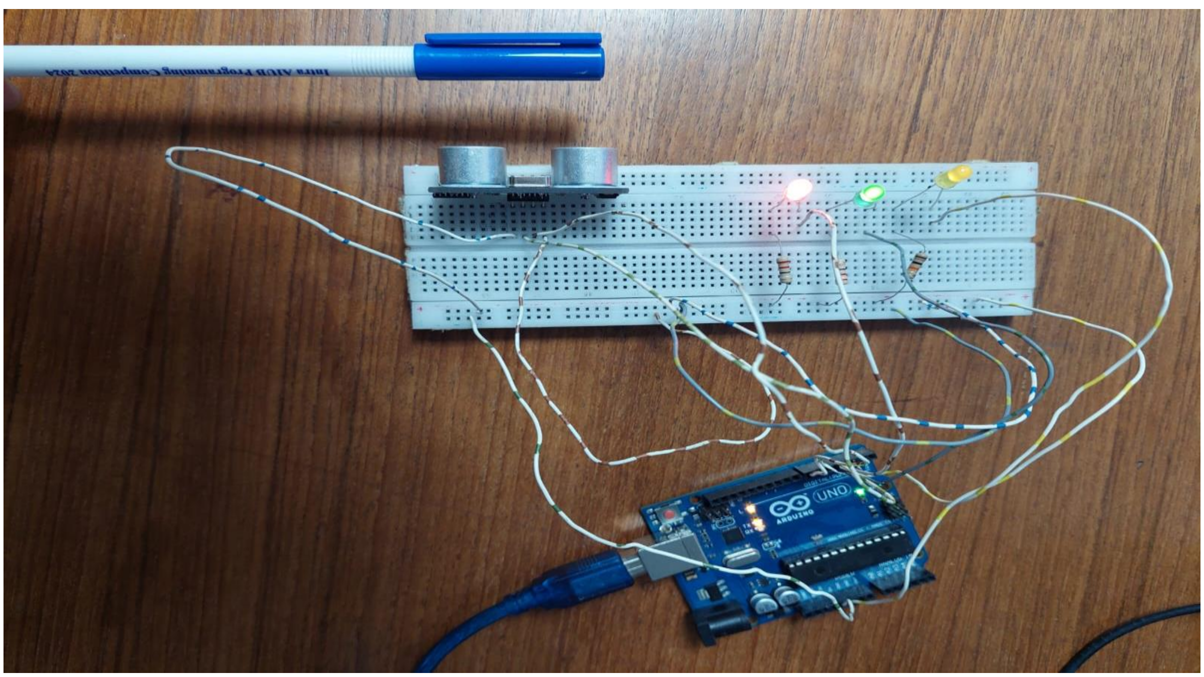

Figure: With obstacles LED is ON

Discussion:

The development of an obstacle detection system using Arduino Uno demonstrates the practical application of microcontroller-based projects. The choice of Arduino Uno for this endeavor is justified by its user-friendly environment, extensive community support, and simplicity in programming. Leveraging the Arduino IDE, participants were able to write code efficiently, utilizing built-in functions and libraries to interface with sensors and peripherals.

The core functionality of the obstacle detection system relies on the ultrasonic sensor (HCSR04) for distance measurement and an LED indicator for obstacle detection. By defining appropriate pins and functions within the Arduino sketch, participants configured the hardware setup and coded the logic for obstacle detection. Key commands such as pinMode() and digitalWrite() were utilized to control the behavior of pins, while additional functions facilitated data processing and conversion.Simulation tools such as Proteus and Tinkercad played a crucial role in validating the functionality of the developed system. Participants could observe the behavior of the system in response to different inputs and scenarios, ensuring its reliability and effectiveness in real-world applications.

Conclusion:

In conclusion, the laboratory experiment provided participants with valuable hands-on experience in developing an obstacle detection system using Arduino microcontrollers. By combining theory with practical implementation, participants gained insight into the process of selecting appropriate components, writing code, and testing the system using simulation tools. The versatility and accessibility of Arduino Uno, coupled with the simplicity of the Arduino IDE, make it an ideal platform for prototyping and developing various electronic projects. Overall, this experiment fosters a deeper understanding of microcontroller-based systems and their potential applications in solving real-world challenges.

Reference(s):

[1] Arduino IDE

[2] Arduino and Proteus Library

[3] Ultrasonic Distance Sensor in Arduino with Tinkercad

Simulation:

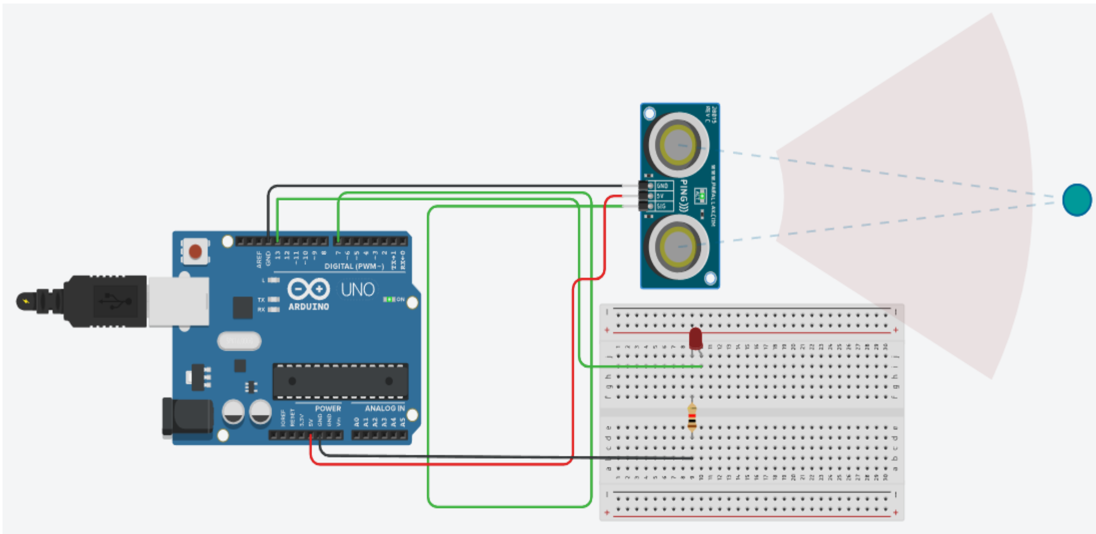

Figure : Without obstacles LED is OFF

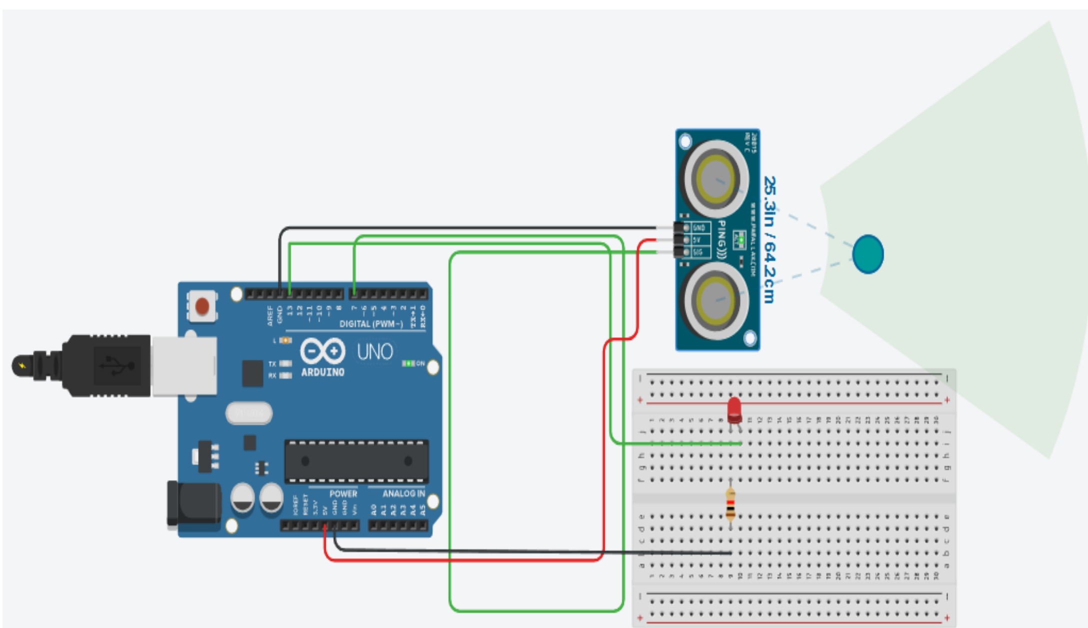

Figure: With obstacles LED is ON

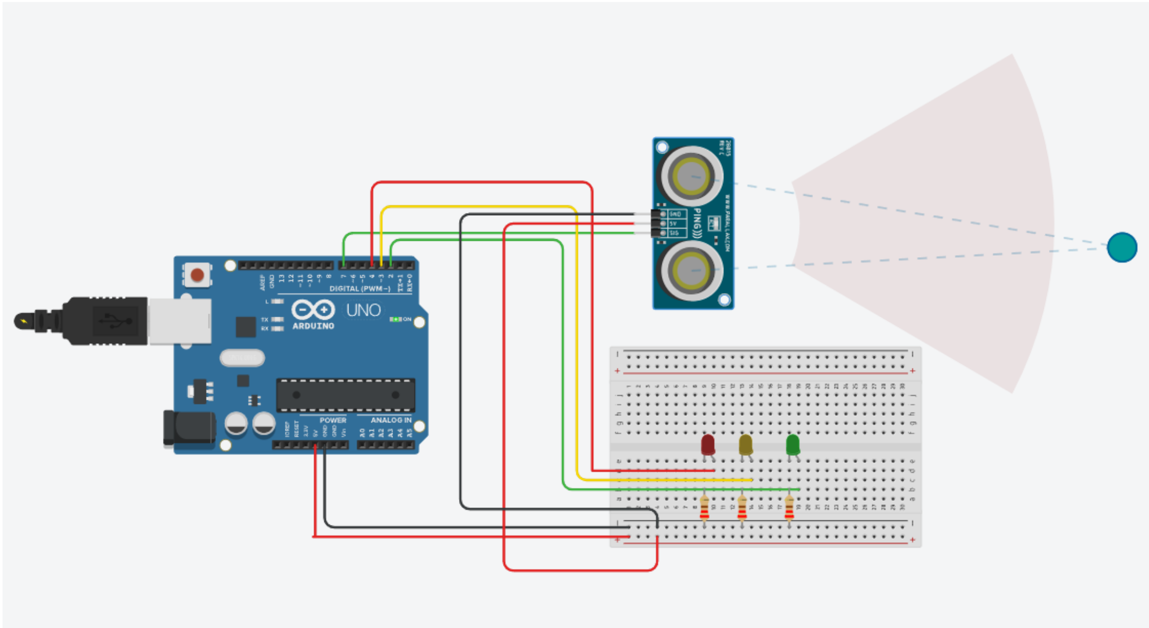

Figure: Without obstacles LED is OFF

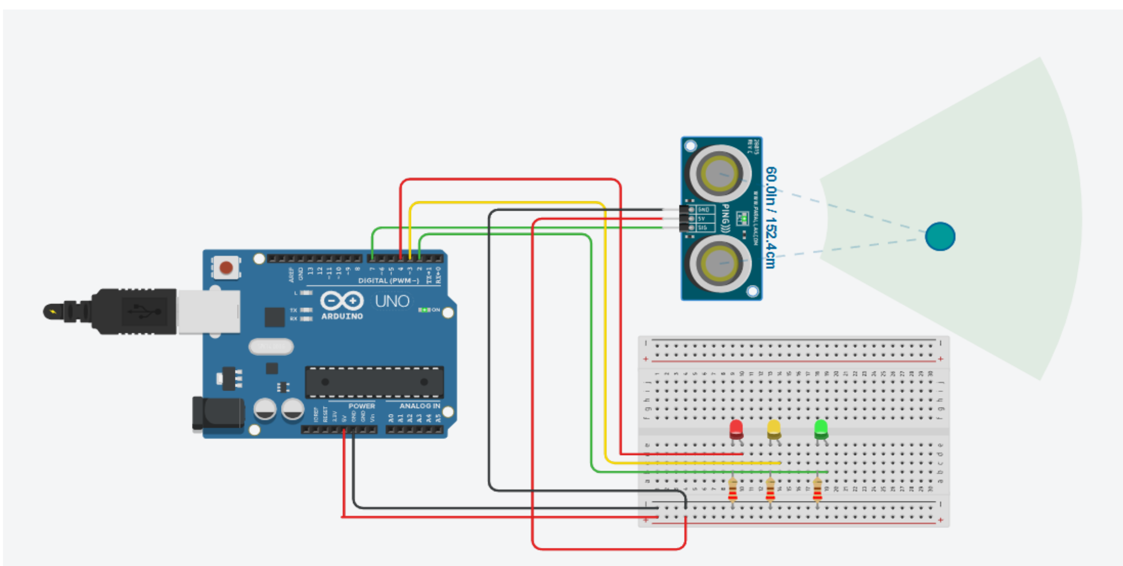

Figure: With obstacles LED is ON