



Arduino Based door Automation System Using Ultrasonic Sensor and Servo Motor

Orji EZ*, Oleka CV, Nduanya UI

Computer Engineering, Enugu State University of Science and Technology, Enugu, Nigeria

Abstract Opening and closing of doors have always been a tedious and boring job, especially in places like; hotels, shopping malls, theaters, etc where a person is always required to open and close the door for visitors. This human involvement can be avoided by automating the process using different sensors like infrared, pressure, ultrasonic, laser etc. In this paper, automatic door control system using Arduino microcontroller was designed. The system combines ultrasonic sensor, servo, and Arduino to achieve the desired goal. When the ultrasonic sensor installed at the entrance of the building detects a person or an object within the range of the sensor, a signal is sent to the Arduino microcontroller which controls the servo motor to automatically open the door. The door remains open until the object goes out of range of the sensor and in turn closes the door automatically. The results clearly show that the system is cheap, effective, and a reliable means of opening and closing doors in places like retail stores, super markets, factories and the like.

Keywords Arduino, Ultrasonic Sensor, Automation, Servo Motor, Arduino IDE, PWM

1. Introduction

The market for automated doors is growing and becoming more specialized. The type of product that fits best with a particular application is determined by frequency of operation, speed of operation required, new versus existing construction, traffic flow and cost.

Automatic doors are a normal feature in many commercial buildings and infrastructures such as shopping centers and airports, as well as in industrial environments such as factories. Although they come in variety of types, including sliding, swing, folding, etc; they all need to conform to the highest standards of safety.

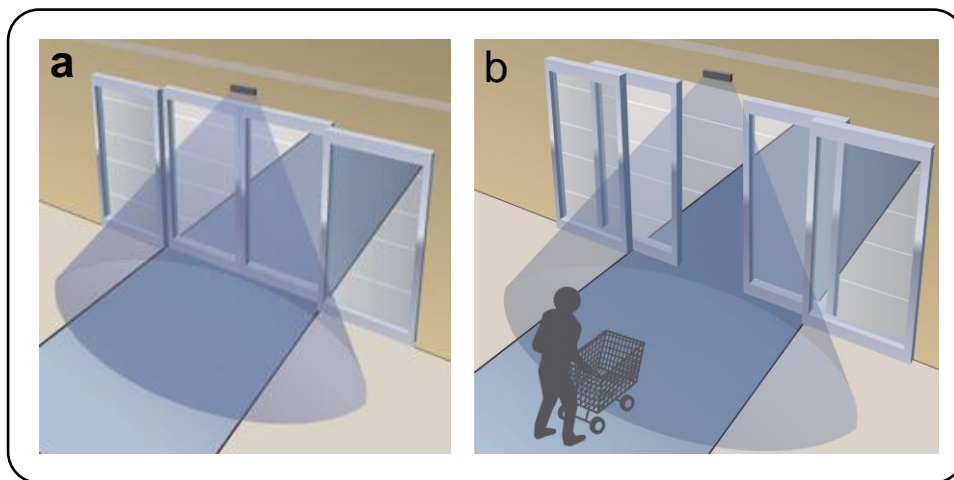


Figure 1: Automatic Door Control System Using Ultrasonic Sensor

In an industrial environment, automatic doors enables the set temperature in cold storage rooms to be maintained by fast opening and closing, leading to considerable energy efficiency and reduction of cost. Automated doors in commercial facilities are the preferred means of access for all users, not only people with disabilities, and are significant aid to accessibility. Moreover, their use minimizes heat or air conditioning loss, maintaining a constant temperature and consequently saving money.

Arduino based door automation system was developed to automate the process of opening and closing of doors. The system uses ultrasonic sensor to detect the presence of an object within the range of the sensor and automatically opens the door or closes the door when the object goes out of range of the sensor as shown in figure 1.

Figure 1 shows two scenarios, first, the door remain closed because no object was detected and second scenario, the door automatically open immediately it detects of an object. In our system, upon detecting an object by the ultrasonic sensor, it sends an information to the Arduino microcontroller who in turn controls the servo motor to open the door and keep it open until the ultrasonic sensor sends another information that the object is no longer in its range of detection. That is, the sensor keeps the detection area under surveillance by emitting ultrasonic waves. The door is left open while people are in the area as shown in figure 1b.

2. Literature Review

Lucky [1] proposed password protected home automation system with automatic door lock which works on the principle of breaking an infrared beam of light, sensed by a photodiode. It consists of transmitting infrared diodes and receiving photo-diodes. The system is to detect whether someone is coming in or not. The photodiodes are connected to comparators, which give a lower output when the beam is broken and high output when transmitting normally. The general operation of the work and performance is dependent on the presence of an object entering through the door and how close the object is to the door. This accomplished detecting an object that is approaching the door but has a drawback which is the distance the infrared can detect an object before it gets to the door.

Mahmood [2] designed an automatic door system using a unique wireless ID by using infrared ray or Bluetooth technology. It consists of a sensing unit, control unit and drive unit to open and close doors at the entrance for a car that has the unique ID. This process is controlled by using Arduino Leonard and programmed with IDE free open source software, that receives the signal code from the car which sends the ID through IR LED or Bluetooth by using a mobile application, decode it. And switch ON the driver that controls the DC motor. Like [1] work, the system detects a car successfully using infrared ray or Bluetooth technology but both technologies have short range detection which is not good because the door might hit the car during opening or closing process.

3. Methodology

This proposed work is about the design of Arduino based door automation system using ultrasonic sensor and servo motor. When a person or an object approaches the door, the HC-SR04 ultrasonic sensor detects the person or objects and sends a signal to the Arduino microcontroller who in turn controls the servo motor to automatically open the door. The door stay open as long as the doorway in not clear and once the doorway is cleared, the ultrasonic sensor sends another information to the microcontroller to close the door until another object is detected.

3.1. Hardware Overview

Detailed description of the hardware components used in carrying out this work will be provided in this subsection.

3.1.1. Ultrasonic Sensor

The ultrasonic sensor emits short and high frequency signal. These signals propagate in the air at the velocity of sound. If there is an object or obstacle on its path, it will bounce back to the module. The ultrasonic sensor consists of a multi vibrator, fixed to the base. The multi vibrator is combination of a resonator and vibrator. The resonator delivers ultrasonic wave generated by the vibration. The ultrasonic sensor actually consists of two parts; the emitter which produces a 40 kHz sound wave and a detector that detects 40 kHz sound wave and



sends electrical signal back to the Arduino microcontroller [3]. The HC-SR04 ultrasonic module used in this project has 4 pins, ground, VCC, trig and echo. The Ground and the VCC pins of the module needs to be connected to the ground and the 5 volts pins on the Arduino board respectively and the trig and echo pins to any digital I/O pin on the Arduino board.

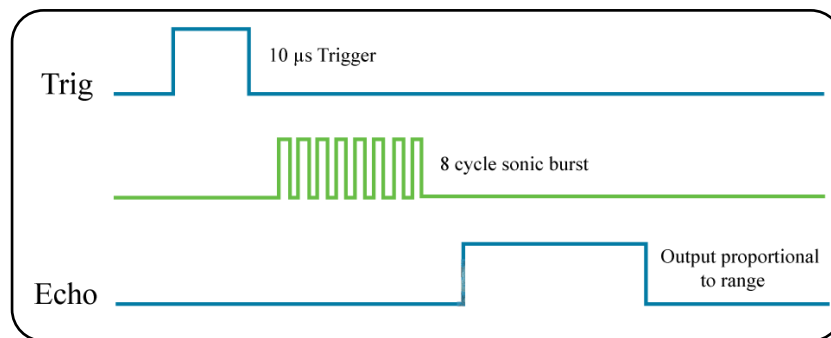


Figure 2: HC-SR04 Timing Diagram

In order to generate the ultrasound you need to set the Trig on a High State for 10 μ s as in figure 2. That will send out an 8 cycle sonic burst which will travel at the speed sound and it will be received in the Echo pin. The Echo pin will output the time in microseconds the sound wave traveled.

Distance Measurement

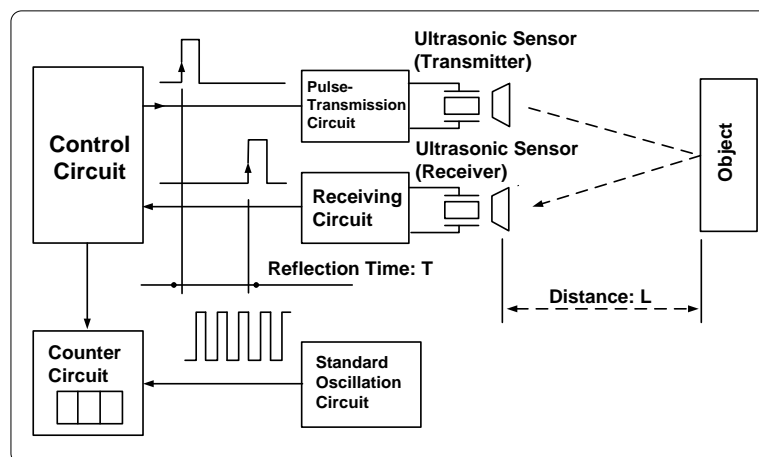


Figure 3: Principles of Measuring Distance

Figure 3 shows the principles of measuring distance and is called the "pulse reflection method" which makes it possible to count the number of reference pulses. This method is used to measure reflection time up to the object between transmitting pulse and receiving pulse of the ultrasonic wave.

The relationship between the distance up to the object L and the reflecting time T is expressed by the following formula:

$$L = S \cdot T/2 \text{ where } S \text{ is the speed of sound.}$$

That is, the distance to the object can be ascertained by measuring the reflection time involved in reaching the object.

For example, if the object is 10cm away from the sensor, and the speed of the sound is 340 m/s or 0.034cm/ μ s the sound wave will need to travel about 294 μ s. But what you will get from the Echo pin will be double that number because the sound wave needs to travel forward and bounce backward. So in order to get the distance in cm we need to multiply the received travel time value from the echo pin by 0.034 and divide it by 2.

Speed of sound = $v = 340 \text{ m/s}$ or $0.034 \text{ cm}/\mu\text{s}$

$$Time = \frac{\text{distance}}{\text{speed}} \quad (1)$$



$$t = \frac{s}{v} = \frac{10}{0.034} = 294\mu s$$

The obtained distance will be twice the actual distance since it gives to and fro distance of the object as per the to and fro time equated to the equation:

$$\text{Distance} = \text{time} * \text{speed} \quad (2)$$

$$l = t * v = 294 * 0.034 = 10\text{cm}$$

Thus the obtained distance divided by 2 gives actual distance of the obstacle.

$$\text{Distance} = \frac{\text{time} * \text{speed}}{2} \quad (3)$$

$$l = \frac{t * v}{2} = \frac{294 * 0.034}{2} = 5\text{cm}$$

Measurement Scenarios

Figure 4 shows three typical ultrasonic sensor measurement scenarios.

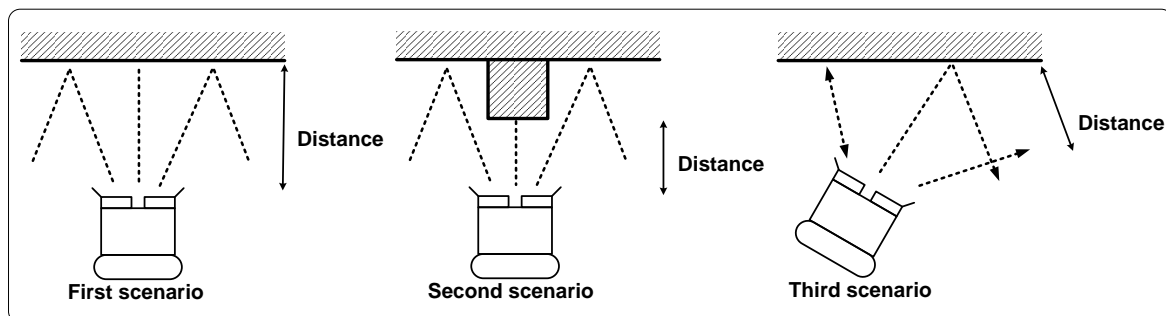


Figure 4: Typical Ultrasonic measurement scenarios

The first scenario in figure 4 will generate a precise measurement because the ultrasound sensor is opposite and perpendicular to the obstacle.

The second scenario will also generate a precise measurement, but will give a “view” of the obstacle located directly opposite the ultrasound sensor.

While the third scenario will generate an inaccurate measurement however, given that it is the left side of the ultrasound sensor that is taking the measurement.

It is essential to properly understand the beam structure for the ultrasound sensor being used if it is intended for map building. This is less true for obstacle avoidance.

3.1.2. Servo Motor

A Servo is a small device that incorporates a three wire DC motor, a gear train, a potentiometer, an integrated circuit, and an output shaft bearing [4]. Of the three wires that stick out from the motor casing, one is for power, one is for ground, and one is a control input line. The shaft of the servo can be positioned to specific angular positions by sending a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft. If the coded signal changes, then the angular position of the shaft changes. A very common use of servos is in radio controlled models like cars, airplanes, robots, and puppets. They are also used in powerful heavy-duty sail boats. Servos are rated for speed and torque.

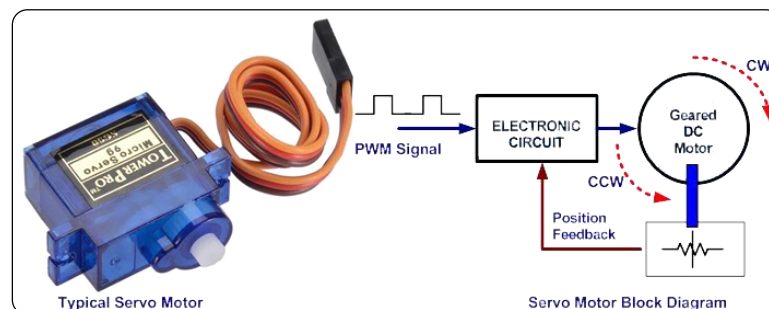


Figure 5: SG90 Micro Servo Motor



The servo motor in figure 5 use pulse width modulation (PWM) signal for controlling the DC motor; unlike normal PWM usually used in ordinary DC motor; this PWM signal is not use for controlling the rotation speed, instead it is use for controlling the motor direction or position [5]. Most servo motor will work well on 50Hz of PWM frequency; this mean the PWM signal should have a period of 20ms. We can vary SG90 Micro servo motor angular rotation in between 0° to 180° angle with PWM signal as shown in figure 5.

Operation of Servo Motors

- It consists of dc motor, gear assembly and feedback control circuitry. PWM signal is used to control the servo motor. It is applied on control signal pin.
- Servo feedback control circuitry contains comparator which compares the control signal (PWM) and potentiometer reference signal to generate error signal which is later amplified and given to the DC motor.
- DC motor shaft is connected to potentiometer shaft (knob) through gear assembly. So rotating DC motor rotates potentiometer, which in term changes potentiometer reference signal given to the comparator.

At some position of shaft, both potentiometer signal and control signal strength match, which produces zero error signal output. Hence rotation continues till comparator output error signal becomes zero and DC motor stops.

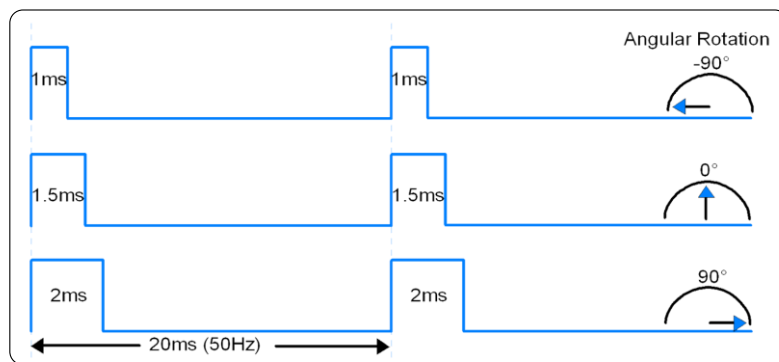


Figure 6: Servo Motor PWM Timing Diagram

Above figure 6 shows angular rotation of servo shaft. It uses PWM of 50Hz frequency with TON variation from 1ms to 2ms. The servo motor rotates 90° in CW (clockwise) and CCW (counter clockwise) direction from its middle position i.e. it gives control over 180° of its rotation.

At $\sim 1\text{ms}$ (5% duty cycle) we get shaft position at -90° (CCW) of its rotation.

At 1.5ms (7.5% duty cycle) we get shaft position at 0° (center) of its rotation.

At $\sim 2\text{ms}$ (10% duty cycle) we get shaft position at $+90^{\circ}$ (CW) of its rotation.

3.1.3. Arduino Board

Arduino is a small microcontroller board with a USB plug to connect to computer. It has number of connection sockets that can be wired up to external electronics, such as motors, relays, light sensors etc. They can either be powered through the USB (universal serial box) connection from the computer or from a 9V battery. They can be controlled from the computer or programmed by the computer and then disconnected and allowed to work independently.



Figure 7: Arduino UNO Board



Arduino UNO- The Arduino Uno is a microcontroller board based on the ATmega 168. It has 14 digital input/output pins (of which 6 can be used as PWM (pulse width modulation) outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP (in-circuit serial programming) 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 a battery to get started [6]. Figure 7 above depicts a typical Arduino UNO board.

3.2. Software Overview

In depth description of the software component used in carrying out this work will also be provided in this subsection.

3.2.1. Arduino IDE

The IDE (Integrated Development Environment) is a special program running on your computer that allows you to write sketches for the Arduino board in a simple language modeled after the processing (www.processing.org) language [7]. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a sketch. Arduino programs are written in C or C++. The Arduino IDE comes with a software library called “Wiring” from the original Wiring project, which makes many common input/output operations much easier.

The basic structure of the Arduino programming language is fairly simple and runs in at least two parts. These two required parts or functions enclose blocks of statements.

```
void setup()
{
    statements;
}
void loop()
    statements;
}
```

Where setup() is the preparation, while loop() is the execution. Both functions are required for the program to work. The setup function should follow the declaration of any variable at the very beginning of the program. It is the first function to run in the program, it runs only once and is used to set pinMode or initialize serial communication. The loop function follows next and includes the code to be executed continuously - reading inputs, triggering outputs, etc. This function is the core of all Arduino program and does the bulk of the work [6].

4. Automatic Door Control System

The key component being utilized in automatic door control system using Arduino is the ultrasonic sensor HC-SR04 and servo motor.

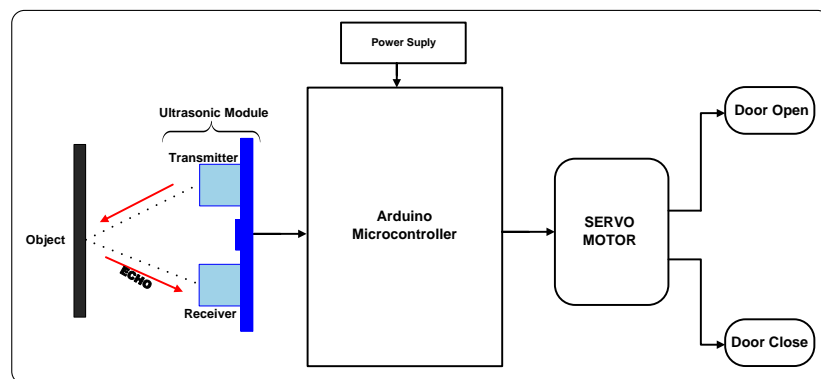


Figure 8: System Block Diagram



The block diagram of the proposed door automation system, which includes opening and closing actions, is depicted in figure 8. When the HC-SR04 ultrasonic distance sensor senses an object within the specified range of detection, a signal is sent to the microcontroller which in turn opens the door accordingly with the help of the servo motor. On the other hand, when there is no object within the specified range, the closing of the door is activated. The door remains open as long as objects are still within the specified range of detection.

4.1. Control Flow

The control flow diagram of the proposed door automation system, which includes opening and closing actions, is depicted in figure 9.

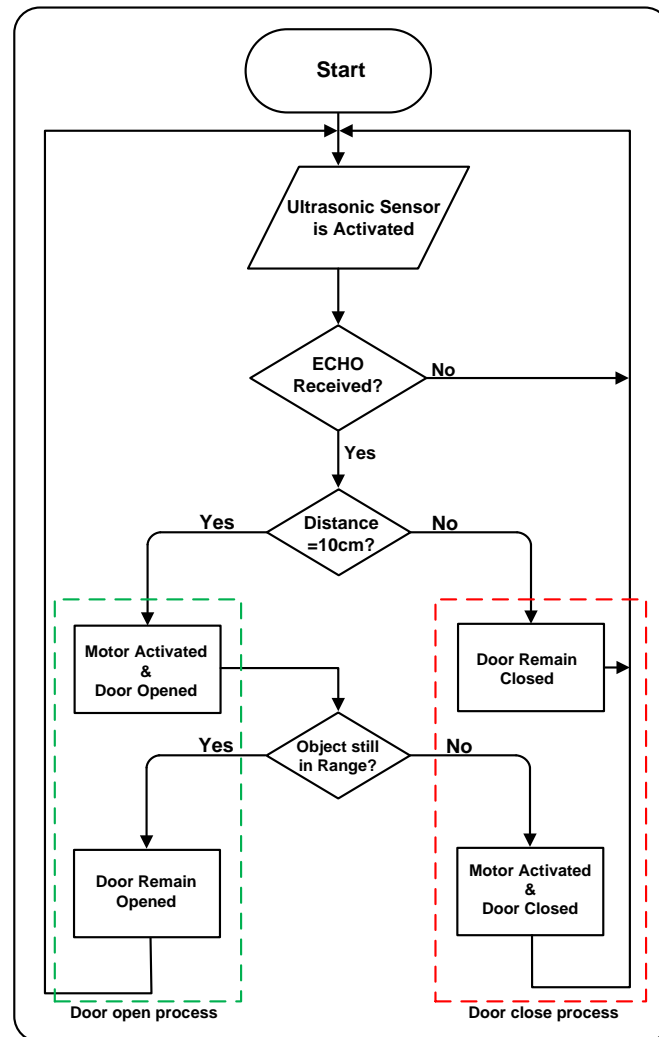


Figure 9: Control Flow Chart

Figure 9 shows control flow of our system. When the ultrasonic sensor is activated, it waits to receive an echo which will be as a result of an object within the specified range of detection (10cm). At the initial state the door was closed assuming no object was detected. Once an object is detected, the servo motor is activated and the door open as a result. The ultrasonic sensor keeps on checking if the detected object is within its range. On the other hand, if the object is out of range, the servo motor is activated again, this time for closing of the door else the door remains opened.

The Arduino based door automation system schematic diagram is shown in figure 10. The schematic has three separate parts; an ultrasonic sensor, a controller, and a servo motor. The ultrasonic sensor detects presences of an object in the detections area and sends a control signal to the microcontroller. The servo motor is there to perform a control action (opening and closing the door) from the microcontroller.



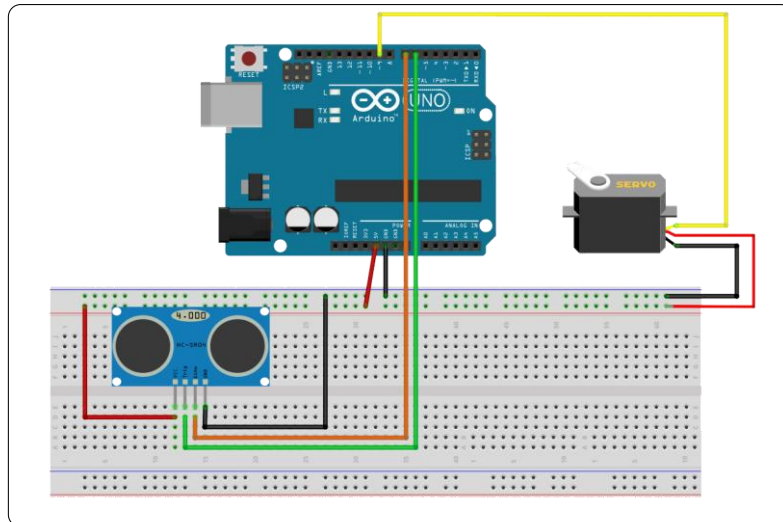


Figure 8: Schematic Diagram of the system

Table 1: Connection of the Ultrasonic Sensor, Servo Motor with Arduino Microcontroller

Ultrasonic Module	Arduino
Trigger Pin	Digital Pin 7
Echo Pin	Digital Pin 8
Vcc Pin	5v
GND Pin	GND
Servo Module	Arduino
Data In Pin	Digital Pin 9
5v Pin	5v
GND Pin	GND

With the aid of the schematic diagram in figure 8, the connection in table 1 was made. It aid in interfacing the ultrasonic sensor, servo motor, and the Arduino microcontroller.

4.1. Results

Working Principle of Arduino based door automation system using ultrasonic sensor and servo motor illustrated in figure 9 is as follows:

If there is no person or object within radar of the ultrasonic sensor (10cm), the door remains closed as shown in figure 9(b). But if there is an object or a person within the ultrasonic radar, the door will automatically open as in figure 9(a).

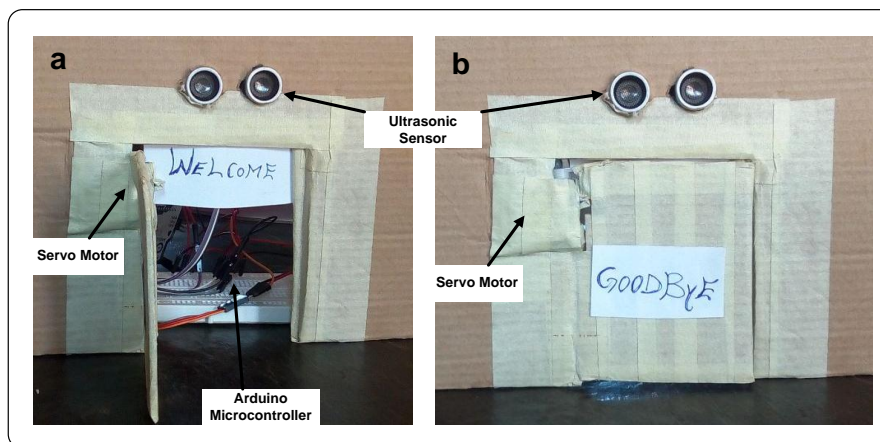


Figure 9: (a) System open state (b) System close state



The door will remain open until the object goes out of the ultrasonic radar, then it will automatically close. The system is setup with one Ultrasonic sensor (in front) for ease purpose, but can be extended by adding another sensor at the back using the same process. This will help to automate entry and exit of our system.

5. Conclusion

This paper presented a prototype of Arduino bases door automation system using ultrasonic sensor and servo motor. The door automation system uses ultrasonic sensor to detect presence of human or an object within its radar and sends a signal to Arduino microcontroller who instructs the servo motor to open the door and keeps it open. Once the object is out of the ultrasonic sensor radar, it sends another signal to the microcontroller for it to instruct the servo motor to close the door automatically.

Since the door is opened only when a person is detected and remains close all other times, it can save a lot of energy in the form air conditioning and can be useful for aged and disabled person.

5.1. Limitations

Although this technique is successful in opening and closing the door when a person or objects are detected, the system is not capable of understanding the type and the intention of the objects. For instance, a puppy or a passing pedestrian may accidentally trigger the door and cause a false opening action. Frequent false action is not only annoying, and results in air conditioning energy waste, but also reduces equipment lifetime. Also the system only serve entry but none was set up exiting using the same door.

Reference

- [1]. Lucky Gautam, Chinoy Sharma, Akshaj Arora, and Pinku yadav, "Developing Infrared Controlled Automated Door System", International Journal of Modern Engineering Research (IJMER) Vol. 3, Issue. 5, Sep - Oct. 2013 pp-2872-2874.
- [2]. Mahmood, S.H., Hassan, O.G., Abass, S.F. and Kwad, A.M, "Auto Opening Door and Car Identification", Journal of Computer and Communications, 2016, 4, 132-141.
- [3]. Dejan Nedelkovski, (2015), "Ultrasonic Sensor HC-SR04 and Arduino Tutorial", Retrieved 25 September 2017, from <http://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>
- [4]. Tesla Institute, (2016), "Servo motor", Retrieved 26 September 2017, from <http://www.tesla-institute.com/index.php/servo-motor>
- [5]. RWB, (2009), "Basic Servo Motor Controlling with Microchip PIC Microcontroller", Retrieved 27 September 2017, from <http://www.ermicro.com/blog/?p=771>
- [6]. Brian W. Evans, (2007). "Arduino Programming Notebook, First Edition", Brian W. Evans.
- [7]. Massimo Banzi, (2012), "Getting started with Arduino", Beijing, O'Reilly.

