DEVELOPMENT OF OBSTACLE AVOIDING FIRE FIGHTING ROBOT

Princewill Chigozie Ene[1], Uche Patrick Okafor[2] Chisom Ifeanyieze[3],

Department of Electrical and Electronic Engineering, Enugu State University of Science and Technology[123]

eneh.princewill.@esut.edu.ng[1], ptuch7777@yahoo.com[2], chisomifeanyieze2010@gmail.com[3]

Corresponding Author: eneh.princewill.@esut.edu.ng[1]

Abstract:

In this paper, an obstacle avoidance for firefighting robot was designed and implemented using a photodiode, ultrasonic sensor, camera, DC motors, water pump, raspberry pi and a wireless remote controller. In achieving this work, the kinematic model of the four-wheel robot was developed first using appropriate mathematical expression. A Raspberry pi with microcontroller inside was programmed using python language in Rasbian software to send, receive and process signals from the interfaced sensors. Omni-directional movement of the robot was achieved using DC motors with rolling wheels attached to each of them. The system control was achieved remotely using transmitter, receiver at 433MHz, HT12 Decoder and Encoder. Simulation and experimental results show that the developed system can avoid obstacles within 20cm of its standpoint. Results also show that the robot at 15cm apart can sense and extinguish fire.

Keywords – Raspberry pi, Firefighting Robot, Python, Obstacle Detection, Sensors.

1.0 Introduction

Many people need fire for different purposes, but they are afraid of the flame of the fire. Fire is a very dangerous element that can consume anything on its path when not controlled. Fire is normally extinguished by fire fighters who take great risks trying to extinguish the flame. Toxic gases are normally released when substances are being burn. These gases can cause health hazards to any living thing and probably death. Fire fighters inhale these gases trying to put out the flame in burning buildings or cars etc. Apart from the toxic gases, electronics or vehicles have tendency of exploding when in flames. When there is an explosion from electronic components or vehicles, human beings exposed to it are injured physically and sometimes can lead to death. It has been observed that fire fighter's response time is dependent on the location of the outbreak and could lead to loss of lives and properties if not responded on time. These flops necessitated the development of a firefighting robot that is not at risk of any health hazard or lose of life.

A robot is a machine, especially one programmable by a computer capable of carrying out a complex series of actions automatically. Robots can be guided by an external control device or the control may be embedded within [1]. The actions performed by the robot depend on the programmer and what the robot is designed to actualize.

Fire extinguishing robots make use of flame sensors to detect when there is fire outbreak. For a fire extinguishing robot to successfully move around its environment without incurring any damage, it needs to avoid obstacles. Obstacle avoidance is a very vital part of mobile robotics; because without it, movement of robots would be restricted or limited by objects along its part [2]. This is important because the robot environment is dynamic and has unpredictable behaviors, i.e. the surrounding does not remain the same and thus the sensors (sensory information) are used to detect the changes in the environment for immediate adaptation.

A sensor is a device, module or subsystem whose main purpose is to detect changes in its environment and send information to other units of the electronics for further processing [3]. Sensors that detect obstacles include; Infrared sensors (IR), ultrasonic sensors, cameras, Passive Infrared Sensor (PIR) among others. The technique of distance measurement using ultrasonic sensor include continuous wave and pulse echo technique, [4]. In the pulse echo method, certain pulses (in form of sound signals) are sent through the transmission medium (air) and it is being reflected by any object at its front. The time taken for the pulse to propagate from transmitter to receiver is proportional to the distance of object.

Currently, there is no fire extinguishing robot available in companies, homes or other establishment in Nigerian. These institutions face the problem of fire at one time or the other. Some of these institutions install fire alarm system and manually operated fire extinguishers. But in case of fire outbreaks in student hostels, offices and banks among others, people are after their lives and most often do not remember the location fire extinguishers are installed. Sometimes the fire extinguisher may expire and become ineffective. Many organizations and institutions especially in Africa are interested in making use of fire extinguishing robot, but have no research work or practical knowledge available for proper design and implementation. Therefore, this study was carried out to develop a prototype fire extinguishing mobile robot incorporated with water pump and flame sensor which can be kept in homes, farms, offices, industries among others to help in automatic detection and extinguishing flame during fire outbreak

2.0 Literature review

A review of some state of the art work on simple mobile robot system were reviewed. Other recent advances in relevant areas were also considered along with well accepted tradition theory and principles associated with the proposed design.

In [4], an algorithm for the operation of the robot with two Direct Current (DC) motors which were powered by a battery was developed. Implementation of intelligent mobile robot based on Microcontroller using Three Ultrasonic Sensors, was presented in [5]. They made use of three ultrasonic sensors which was simulated in the Proteus environment with an Arduino. The sensor measures the distance between an obstacle and the sensor. Light Emitting Diode (LED) and Light Dependent Resistance (LDR) were used as object detecting sensor in [6].

For this work, Raspberry pi with ultrasonic sensor was used for obstacle detection. The robot uses a photodiode for fire detection and was made to be controlled through an external device. Survey of its environment was achieved through programming of the camera interfaced to the controller.

3.0 Methodology

Modular methodological approach was adopted for this work. Each unit of the system was designed before integrating them into a whole.

The complete system circuit was debugged to obtain the correct working design for its physical operation.

The system block diagram is as shown in figure 1. Each block is actually a circuit unit that does a specific work. The kinematic and dynamic models of a four wheeled mobile robot were studied to understand its behavior.

The system was designed using 5 functional blocks; the raspberry pi microcontroller, ultrasonic sensor, flame sensor, motor driver, DC water pump, and the power supply.

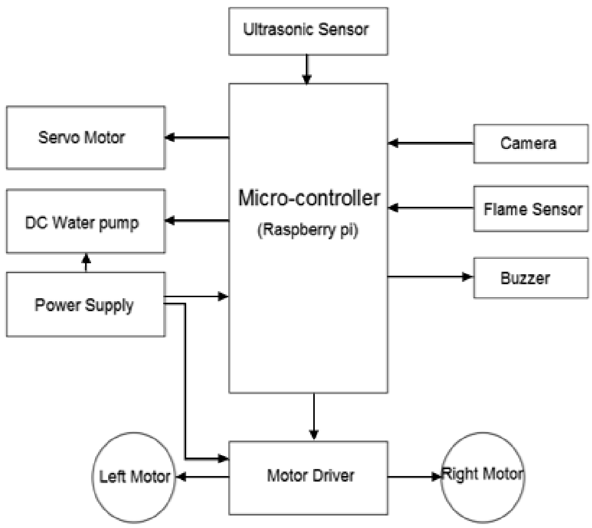


Figure 1: Block diagram of the fire extinguishing robot.

The Raspberry pi mini computer controls every other attached unit in the system. The DC power supply sends the required operating voltages to the various unit of the robot. The servo motor rotates the water spraying nozzle that runs through the DC water pump when the action of putting off fire with the fluid is activated. The Ultrasonic sensor senses the presence of an obstacle on the path of the robot while the flame sensor observes the presence of burning flame with the coverage view of the sensor. Motor driver was used to drive the motor. The driver takes a low current control signal from the microcontroller and provides a higher current signal, thus acting like a current amplifier.

The sequence of operation of the system is developed according to the flow chart of figure 2.

From the flow chart, the five possible states of the robot will be identified after the initialization of the controller. Thereafter, the presence of flame is activated followed by the measuring of 15 metres distance from obstacle in the front of the robot. If any flame is detected, the robot stops and activate the alarm, servo motor and water pump. But if the distance is less than 15 metres, meaning there is an obstacle, the robot switches to obstacle avoiding mode.

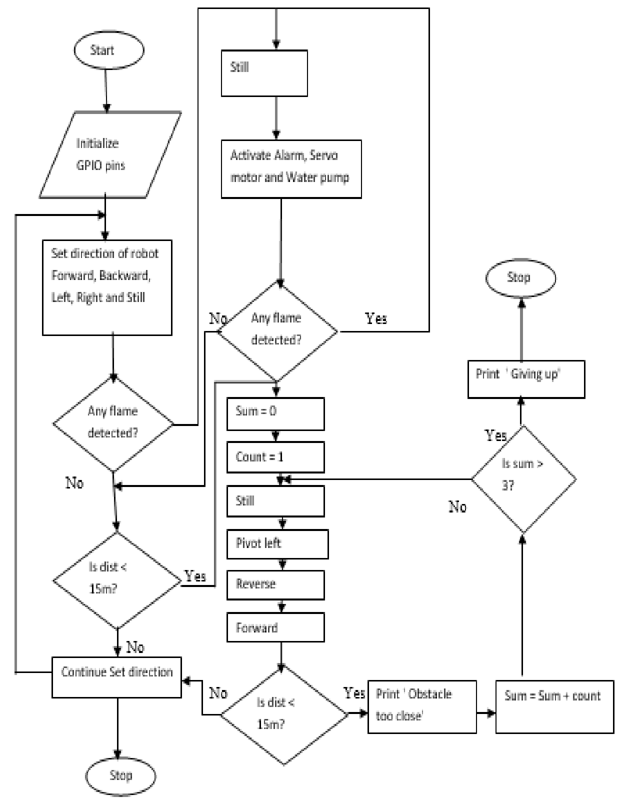


Figure 2: Flow Chart for the Fire Fighting Robot Operation.

3.1 System Design and analysis

a) Kinematic Model of a Four-Wheel Robot

The mobile robot is represented in a global coordinate frame in the x and y axis as shown in figure 3.

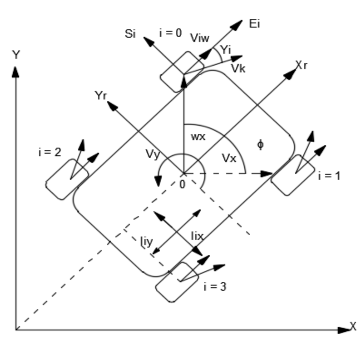


Figure 3: Four wheel robot in a global coordinate frame [7].

Two coordinate describes the position and orientation of the mobile robot in its environment;

1. **Inertial coordinate system:** This coordinate system is a global frame that is fixed in the environment or plane in which the mobile robot will move in. This frame is denoted as $\{X, Y\}$.

2. **Robot coordinate system:** This frame is the frame attached to the mobile robot and thus moving with it. This frame is denoted as $\{X_r, Y_r\}$.

The robot position and orientation in the Inertial frame can be defined as

$$q^I = \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \theta \end{bmatrix} \tag{1}$$

Where the X and Y are the coordinate and θ is the heading direction angle.

It should be noted that the mapping between the two frames are different. The position of any point on the robot can be defined in the robot frame and the inertial frame as shown below;

Let $X^r = \begin{bmatrix} x^r \\ y^r \\ \theta^r \end{bmatrix}$, and $X^I = \begin{bmatrix} x^I \\ y^I \\ \theta^I \end{bmatrix}$ be the coordinates of the given point in the robot frame and inertial frame respectively.

Then, the two coordinates are related by the following transformation:

$$X^I = R(\theta)X^r \tag{2}$$

Where R(θ) is the orthogonal rotation matrix.

$$\mathrm{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

This transformation enables the handling of motion between frames.

$$\dot{X}^I = R(\theta)\dot{X}^r \tag{4}$$

The motion of a mobile robot is characterized by two non-holonomic constraint equations that are obtained by two main assumptions:

1. **No lateral slip motion:** This constraint means that the robot can move only in a curved motion (forward and backward) but not sideward. This means no movement in the y-axis, and the velocity of the center-point (O) is zero along the lateral axis:

$$y^r = 0 \tag{5}$$

Using the orthogonal rotation matrix R(θ), the velocity in the inertial frame gives

$$-x_a \sin\theta + y_a \cos\theta = 0 \tag{6}$$

2. **The pure rolling constraint:** This constraint represents the fact that each wheel maintains a contact point (P) with the ground as shown in figure 4. There is no slipping of the wheel in its longitudinal axis (x_r) and no skidding in its orthogonal axis (y_r).

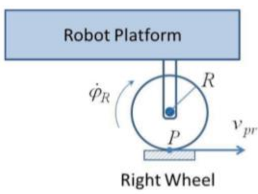


Figure 4: Rolling motion constraint [8].

The velocities of the contact points in the robot frame are related to the wheel velocities by:

$$\begin{cases} v_{pR} = R\dot{\varphi}_R \\ v_{pL} = R\dot{\varphi}_L \end{cases} \tag{7}$$

The velocities in the inertial frame can be expressed as a function of the velocities of the robot center-point (O) as [8];

$$\begin{cases} \dot{x}_{pR} = \dot{x}_a + L\theta\cos\theta \\ \dot{y}_{pR} = \dot{y}_a + L\theta\sin\theta \end{cases} \tag{8}$$

$$\begin{cases} \dot{x}_{pL} = \dot{x}_a + L\theta\cos\theta \\ \dot{y}_{pL} = \dot{y}_a + L\theta\sin\theta \end{cases} \tag{9}$$

Using the rotation matrix R(θ), the rolling constraint equations are formulated as follows:

$$\dot{x}_{pR} cos\theta + \dot{y}_{pR} sin\theta = R\dot{\varphi}_R$$
$$\dot{x}_{pL} cos\theta + \dot{y}_{pL} sin\theta = R\dot{\varphi}_L \quad (10)$$

Using the contact point's velocities and substituting in (x, y), the three constraint equations are written in a matrix form as shown below:

$$A(q)\dot{q} = 0 \quad (11)$$

Where

$$A(q) = \begin{bmatrix} -sin\theta & cos\theta & 0 & 0 & 0 \\ cos\theta & sin\theta L & -R & 0 & 0 \\ cos\theta & sin\theta & -L & 0 & -R \end{bmatrix} \quad (12)$$

and

$$\dot{q} = [\dot{x}_a \dot{y}_a \dot{\theta} \dot{\varphi}_R \dot{\varphi}_L]^T \quad (13)$$

Then, the right and left wheel's velocity are;

$$\begin{cases} v_R = R\dot{\varphi}_R \\ v_L = R\dot{\varphi}_L \end{cases} \quad (14)$$

The linear velocity of each driving wheel in the Robot frame is the average of the linear velocities of the four wheels

$$v = \frac{v_R + v_L}{4} = R\frac{\dot{\varphi}_R + \dot{\varphi}_L}{4} \quad (15)$$

and the angular velocity of the mobile robot is

$$\omega = \frac{v_R - v_L}{4L} \quad (16)$$

The mobile robots' velocities in the robot frame can now be represented in terms of the center point 0 velocities in the robot frame as follows:

$$\dot{x}_a^r = R\frac{\dot{\varphi}_R + \dot{\varphi}_L}{4} \quad (17)$$

$$\dot{y}_a^r = 0 \quad (18)$$

$$\dot{\theta} = \omega = R\frac{\dot{\varphi}_R + \dot{\varphi}_L}{4L} \quad (19)$$

Thus

$$\begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{4} & \frac{R}{4} \\ 0 & 0 \\ \frac{R}{4L} & \frac{R}{4L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \quad (20)$$

The mobile robot velocities can be obtained also in the inertial frame as follows:

$$\dot{q}^I = \begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{4} cos\theta & \frac{R}{4} cos\theta \\ \frac{R}{4} sin\theta & \frac{R}{4} sin\theta \\ \frac{R}{4L} & \frac{R}{4L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \quad (21)$$

Equation 21 represents the forward kinematic model of a four wheeled drive mobile robot.

b) Microcontroller (Raspberry PI)

The raspberry pi is a mini computer that has 40 General Purpose Input and Output pins (GPIO) for sending out and accepting signals as shown in figure 5. The raspberry pi contains a port for a camera, making it easy for the connection of a camera. The pin configuration is of the raspberry pi is shown in figure 6.

Figure 5: A Raspberry Pi (Model 3B)

Figure 6: Raspberry Pi (Model 3B) Pin layout.

The Raspberry pi was used on Linux operating system with Rasbian pre-installed in it. Rasbian is a Debian-based computer operating system for Raspberry Pi. The version of Rasbian used is Debian version 9 (Stretch). The raspberry pi also uses a Linux kernel version of 4.14. The Raspberry pi accepts an input current of 25mA (maximum), 4mA (minimum) and a voltage of 3.3v. The Raspberry pi has an HDMI output terminal for viewing both the command line interface and the graphical user interface. The command line terminal icon was used in writing the program.

A new directory was created using the command "cd robotics" where the code contained in python script was saved. The program was executed after the script has been written and saved.

c) Camera Setup

A mini camera was inserted into the Raspberry pi camera's port. The environment was viewed after running the subprogram for video streaming through the command line. The camera was viewed from another computer by enabling Secure Shell (SSH), Virtual Network Computing) VNC on the Raspberry pi configuration.
The VNC viewer and VLC player were installed on the other Computer which was connected to a Wi-Fi network. The Raspberry pi and computer must be connected to the same network before video streaming and control of the Raspberry pi can be achieved. The Internet Protocol (IP) address of the Raspberry pi was obtained. The IP address, user name and password were provided before using the VNC viewer software.

d) DC Motor Driver

The DC motor driver has two PNP transistors and two NPN transistors. The transistors control the movement of the motor either in clockwise or anti-clockwise direction. In the enable pins, either a True (3.3v) or False (0v) is sent to the transistors. When a positive pulse is sent to an NPN transistor it will switch and allow the flow of current from the collector to emitter, while nothing happens to it when a negative pulse is sent. In the same way, when a negative pulse is sent to PNP transistor, it will bias and allow flow of current from collector to emitter and when it is a positive pulse, nothing happens.
The arrangement of the transistors is in such a way that when a positive pulse is sent to enable 1 and a negative pulse to the enable 2, transistors Q3 and Q2 will bias. Q3 will connect the motor to ground while Q2 will connect the motor to a power supply and this will cause the motor to rotate in an anti-clockwise direction and vice versa.

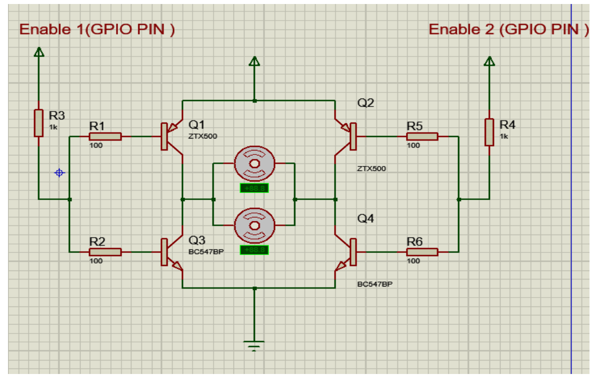


Figure 7: Circuit diagram of the DC motor driver

e) Flame Sensor

A photodiode that received and converted infrared rays electrical signal was used as the flames sensor. The photodiode has three output pins (V_{cc}, Ground and Signal pin). The flame sensor sends 3.3volts to the Raspberry pi through a GPIO pins that received and gives out 3.3volts. A current limiting resistor is connected in series from the Raspberry pi to the flame sensor in other to limit the current of the signal sent. This was done, because a photodiode has tendency of sensing IR rays that radiate out from human. The sun also releases a small amount of infrared rays and to prevent the flame sensor sending false signals to the Raspberry pi constantly, a current limiting resistor was used. The flame sensors circuit diagram is shown in figure 8.

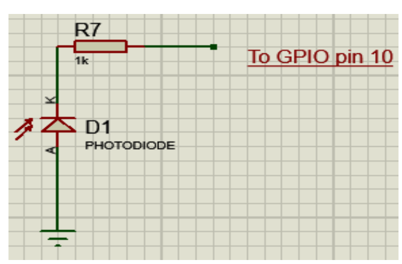


Figure 8: Flame Sensor Schematic.

f) Ultrasonic Sensor

An ultrasonic sensor sends out ultrasonic waves in order to measure distance between itself and object at its front. The sensor has a trigger pin which sends signals for the emission of ultrasonic waves and an echo pin which receives the reflected waves and sends signal to a microcontroller. Distance is usually measured by measuring the time between emission and reception as shown in figure 9.

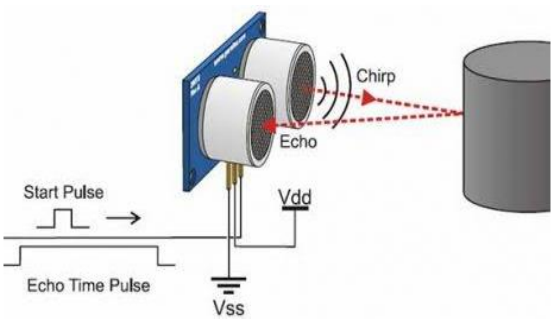


Figure 9: Distance Measurement with Ultrasonic Sensor

$$Object\ distance\ (d) = Speed\ (c) \times Time(t)$$

The time it takes to and fro increases the distance by 2, therefore;

$$2d = tc$$

$$d = \frac{1}{2}tc \tag{22}$$

Where d is the distance, $t$is the time between emission and reception and c is the speed of sound (sonic speed [343m/s])

The 5volts from ultrasonic sensor was scaled down to 3.3volts before getting to the controller using voltage division method as shown in figure 10.

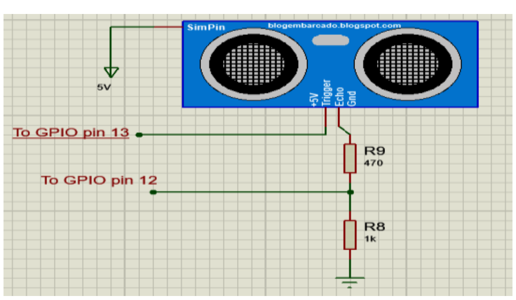


Figure 10: Ultrasonic Sensor Schematic.

The output voltage (Echo) connected to GPIO pin 12 of raspberry pi is;

$$V_{out} = V_{in}\frac{R8}{R9 + R8} \tag{23}$$

Where R2 = 1kΩ resistor and R1 = 470Ω resistor, V_{out} is 3.4volt.

g) DC Water Pump

A separate power source was used to power the DC water pump since the current requirement is more than the controller can source. A driver circuit was made with a transistor and a relay to drive the DC water pump when the robot senses fire.

Since the output voltage from a Raspberry pi is 3.3volts, a resistor of 1kΩ was used to connect to the base of the transistor. The resistor was used to produce a current of (3.3/1000) 0.0033A (3.3mA) which switches a transistor. The collector of the transistor is then connected to one end of the coil as shown in the figure 11.

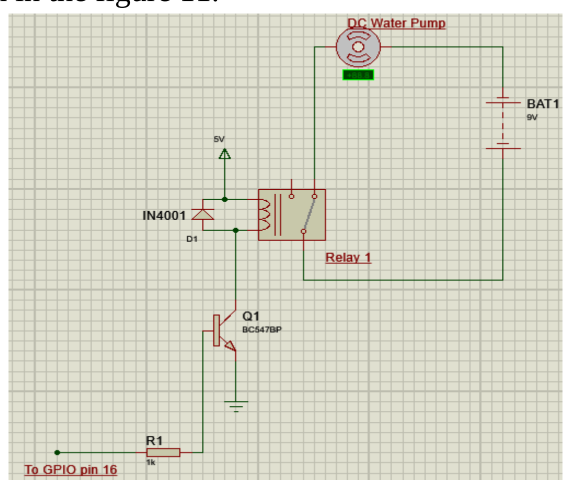


Figure 11: DC water pumping block schematic

A 6volts relay which was operated by a 5V power supply from the 5V pin of the Raspberry pi was used. The DC water pump is then connected to a battery of 12V when the transistor is switched by the signal from pin 16 of the Raspberry pi.

The servo motor makes use of pulse width modulation (PWM) to rotate at a particular angle. The servo motor was made to rotate through 45^o to

90^o using PWM in order to cover the required area. Duty cycle was used to calculate the angle of rotation. The frequency was set at 50Hz which gives 2ms duty cycle. At this duty cycle, the servo motor is at 180^o. At 1ms the servo motor is at 0^o and at 1.5ms the servo motor is at 90^o.

$$Duty\ Cycle = \frac{Pulse\ Width}{Period} \tag{24}$$

and

$$Period = \frac{1}{Frequency} \tag{25}$$

6.25ms in the code to make it rotate an angle of 45^o.

h) Wireless Communication Unit

Remote control module was achieved using a transmitter, receiver, HT12 Decoder and HT12 Encoder (HT12D/E).

The HT12D/E communicates with their input and output pins through a transmitter and receiver. The transmitter transmits at a frequency of 433MHz. From figure 12, the analogue pins of the encoder are all grounded while the data pins (E1 to E4) are connected to a preset switch.

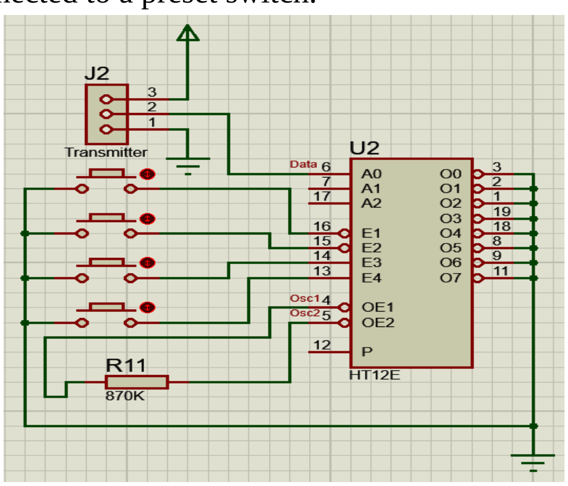


Figure 12: Circuit diagram of the remote transmitter with encoder

A $870K\Omega$ resistor was connected in series with oscillator 1 for signal transmission at one fixed frequency. The decoder frequency was made 50times the frequency of the encoder so that there will be true communication. Therefore, the decoder frequency was 187.5 KHz.

The voltage division rule was used to step down the voltage from 5volts of the output pins of the decoder to 3.3volts for the Raspberry pi. This was done by connecting two resistors (470Ω and 1000Ω) in series, and a wire connected between them. The circuit diagram for the receiver and decoder is shown in figure 13:

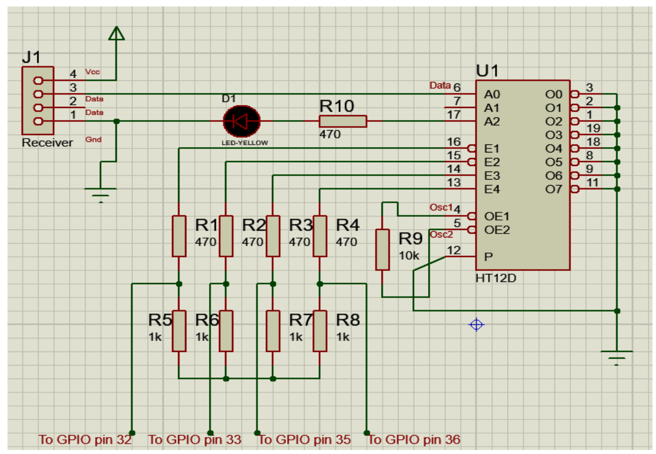


Figure 13: Circuit diagram of the remote receiver with decoder.

The analogue pins were also grounded. The oscillator resistor was chosen to be $39K\Omega$ which was traced using the operation graph.

g) Battery Charger

A simple battery circuit shown in figure 14 was used for charging the 12volts battery which supplies DC power to the water pump.

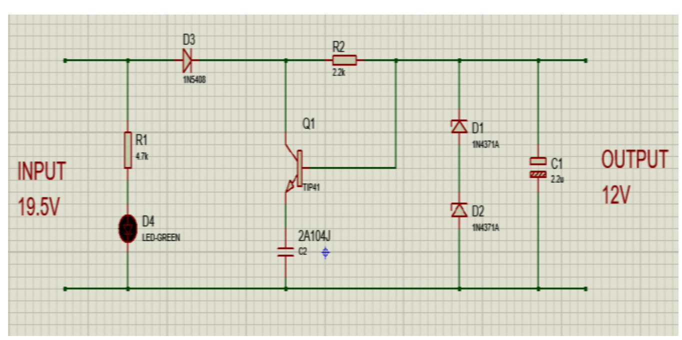


Figure 14: 12volts Battery Charger.

Two zener diodes and a transistor were used to achieve a constant 12V supply.

3.2 Implementation

Before implementation, instruction codes were developed for different functions using python programming language in the Rasbian Command Line Interface (CLI). Some of the sections were simulated in the Proteus and MATLAB environment. The system circuit was debugged to obtain the correct working design for its physical operation.

The ultrasonic sensor was programmed and tested to check if it gives the accurate distance measurement. The predetermined distance before avoiding an obstacle is 20cm. The echo was connected to pin 16 and the trigger to pin 12. The ultrasonic sensor was made to check distance every 10millisecs for accuracy in obstacle avoidance operation.

The flame sensor and buzzer work simultaneously in the sense that once the robot senses a flame, the buzzer is triggered and so does the alarm. The flame senor was connected to pin 18 of the Raspberry pi while the buzzer was to pin 22.

The DC water pump draws too much current and therefore may damage the Raspberry pi. This was why a relay was used as a switch to connect the DC water pump to a power source. The external battery can power the DC water pump for at least 10minutes. The servo motor's trigger pin was connected directly to the Raspberry pi pin 31, while the trigger pin for the water pump was on pin 29. After the connections were done with jumper wires, the right program was written and executed based on different functions of the system.

Finally, the camera was interfaced for image capturing and area surveillance.

4.0 Results and discussions

A binary occupancy grid was created in the MATLAB environment, using the path finding code, the robot was simulated with its sensor (Ultrasonic sensor) activated. Figures 15 to figure 17 show the simulation results of the robot moving through a desired path to three different destinations. Figure 15 shows the robot moving to a destination located at point (12, 10) of the x and y axis of the occupancy grid. Figure 16 shows the robot moving to a destination located at point (6, 1), while Figure 17 shows the robot moving to a destination located at point (2, 10). In Figure 17, the robot was blocked by an obstacle (wall), it took another direction after the obstacle was detected.

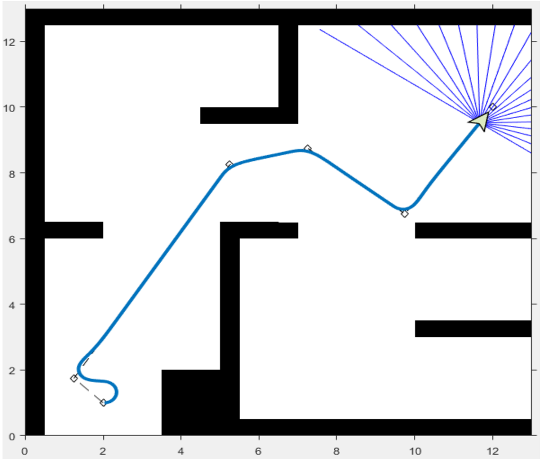


Figure 15: Robot navigation to point (12, 10)

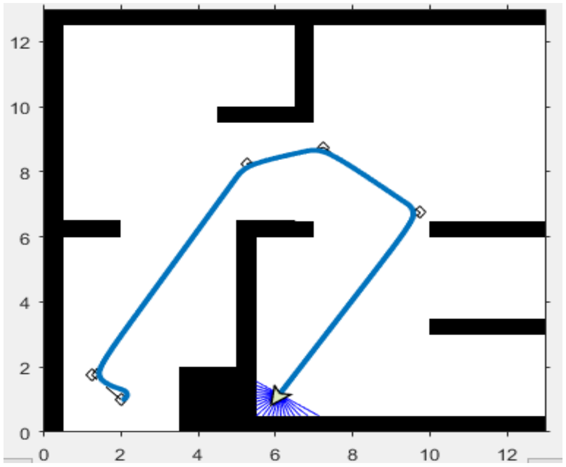
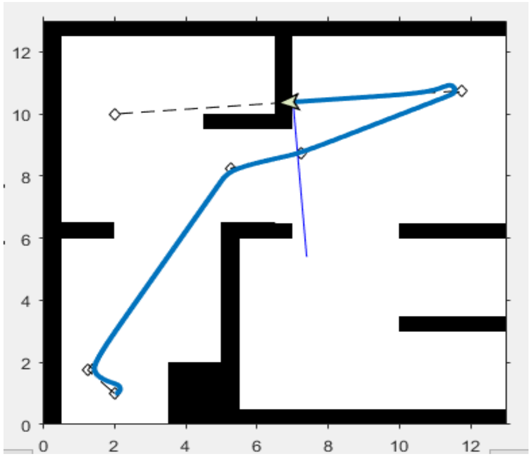


Figure 16: Robot navigation to point (6, 1)



(c)

Figure 17: Robot simulation to point (2, 10).

After programing the Raspberry pi to control individual wheels, the directional movement achieved were forward, backward, left and right with pivot (rotating at a point) as shown in figure 18.

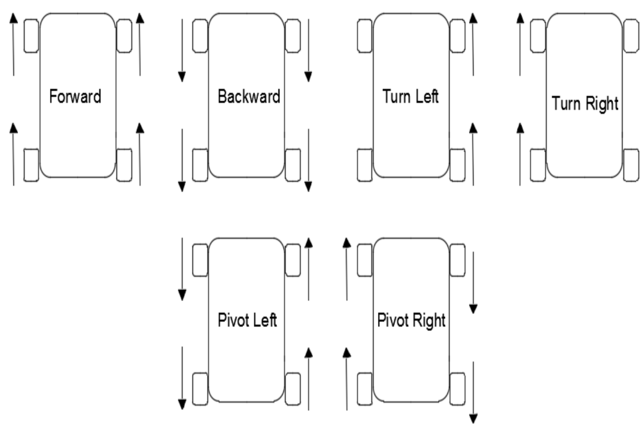


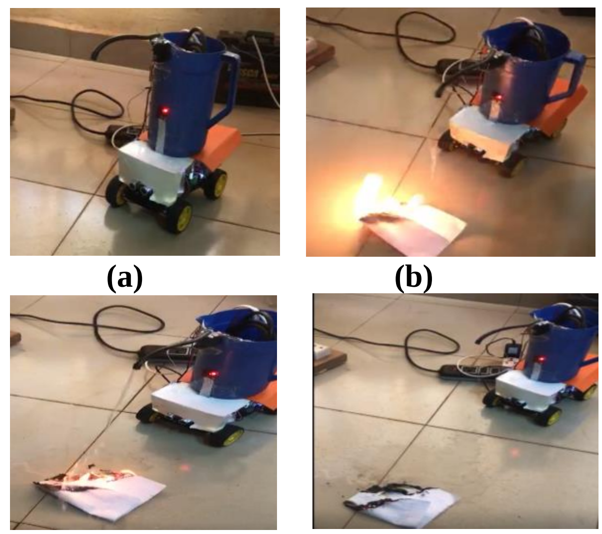
Figure 18: Directions achieved by the mobile robot.

The logic table for the wheels is shown in table 1 with False as 0volts and True as 5volts.

Table 1: Robot wheel truth table.

Direction	Left Wheels		Right Wheel	
Pins	7	11	13	15
Forward	True	False	True	False
Reverse	False	True	False	True
Turn left	False	False	True	False
Turn Right	True	False	False	False
Pivot left	False	True	True	False
Pivot Right	True	False	False	True

The entire system was tested as shown in figure 14. In figure 14a, the robot was powered and placed on a plane floor with no fire in its front and made to move in all four directions. In figure 14b, a paper was set on fire and placed on the floor within 15cm away from the robot, and then the robot was made to move towards the flame. Immediately it sensed the flame, the pump activated with the servo motor and water sprinkled as shown in figure 14c. The pump kept on pumping water out till the fire went off as shown in figure 14c – 14d.



(a) **(b)** **(c)** **(d)**

Figure 14: (a) The system is powered ON, (b) A paper is lit on fire, (c) The robot detects the flame and activates the water pump, (d) The flame has been extinguished.

5.0 Conclusion

An obstacle and firefighting robot that can be used to put out fire at homes and offices was successfully designed and implement using Raspberry pi, ultrasonic sensor, camera, DC motor, water pump and remote controller as major components. The developed system was tested at two instances to determine its responsiveness. At the first instance, burning flames were used. The robot was able to sense the flames and activated its water pump immediately and the fire was put out. At the second instance, an obstacle was placed in the path of the robot within 20cm of its standpoint. It was observed that the robot was able to avoid the obstacle without collision.

There is need for the robot to move autonomously while moving around its environment sensing for presence of fire. Therefore, it is paramount that more studies be instigated in this area.

References

[1] D. Halperin, L. E. Kavraki, and K. Solovey, "Robotics," in *Handbook of Discrete and Computational Geometry, Third Edition*, 2017.

[2] J. Suresh, "Fire-fighting robot," in *ICCIDS 2017 - International Conference on Computational Intelligence in Data Science, Proceedings*, 2018.

[3] M. M. Canteli, "Sensors," in *Handbook of Automotive Power Electronics and Motor Drives*, 2017.

[4] A. Raghav, P. Mishra, P. Verma, K. S. Randhawa, and T. Thakur, "Obstacle Avoiding Robot," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, 2017.

[5] X. T. Le, S. B. Oh, W. S. Lee, C. S. No, and S. H. Han, "Design of intelligent mobile robot system based on ultrasonic sensors," in *ICCAS 2007 - International Conference on Control, Automation and Systems*, 2007.

[6] M. Becker, C. M. Dantas, and W. P. Macedo, "Obstacle avoidance procedure for mobile robots," *ABCM Symp. Ser. Mechatronics*, 2006.

[7] M. Lauria *et al.*, "Design and control of a four steered wheeled mobile robot," in *IECON Proceedings (Industrial Electronics Conference)*, 2006.

[8] R. D. Ahmad Abu Hatab, "Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework," *Adv. Robot. Autom.*, 2013.