Security Feature Using Ultrasonic Sensor For Detection

Ashna Khemani, Iniyan Joseph, Sidrah Ahmed

Advisor: Nicholas Papano

Aspiring Scholars Directed Research Program, 43505 Mission Blvd, Fremont, CA 94539

KEYWORDS: Security, Raspberry Pi, Ultrasonic Sensor

ABSTRACT: In this project, a security device using an ultrasonic sensor was made that causes a PVC pipe to drop on the person who has crossed the path of the sensor via motor. It uses basic Python to control the motor and receive data from the ultrasonic sensor through the GPIO (general-purpose input/output) pins on a Raspberry Pi 3. An ultrasonic sensor will be able to sense anything moving in front of it and will send a "True" signal to the Raspberry Pi. In turn, the Pi will trigger the motor to run forward for (x) seconds, releasing the PVC pipe, then, after a pause, run in the other direction to rewind a fishing line attached from the motor shaft to the PVC pipe, thus bringing the pipe up, and resetting the contraption. This project is meant to teach basic usage of GPIO pins, circuitry using breadboards, effective engineering, and coding in Python.

INTRODUCTION

The end goal was to build a security system that will obstruct a person's path as they walk through a doorway where the system is set up. Currently, there are sensing technologies such as laser sensors and pressure plates. A laser sensor emits a beam of light aimed at a photosensor.[1] If an object crosses the laser beam, the light won't shine on the photosensor, so it will tell the computer that it has detected something.[2] In this project, we used ultrasonic sensors, which uses transducers to send out an ultrasonic signal, receive it, and send the data back to the computer.[3] It is similar to the echolocation used by bats and dolphins.[4] It measures how long it took for the "pulse" to return, and will gauge the distance using that number.[4] The sensor inputs were used to move a motor.

There are hundreds of computers out there than can run hundreds of processes, but a computer with GPIO pins would be extremely helpful as we would be mainly giving inputs and outputs through external devices (sensors and motors).[4] We therefore decided to use a Raspberry Pi 3 since it had a good amount of

GPIO pins as well as other standard and necessary ports, such as HDMI and USB.[7]

METHODS

There were several key components in the system: The first necessary item was a Raspberry Pi to control the motor and receive data from the ultrasonic sensor. The motor needed another power source to function as it went over the amount possible to send via GPIO pins. An ultrasonic sensor was required as it was chosen over a pressure plate or a laser trigger as it was used as a one-part component, unlike the laser sensor, and was more durable than the pressure plate; it would be easier and more efficient to work with. It also had the advantage of being easily concealed and did not require the same level of precision on where the user has stepped.

Setting Up Raspberry Pi

The appropriate OS for the Raspberry Pi would first need to be downloaded. Raspbian Buster With Desktop And Recommended Software, which included various IDEs and code editors, was downloaded to a Macintosh as an NDIF disk image.[8] After that, a microSD card was mounted onto the Mac using an SD adapter and we loaded the opened disk image to the SD using Etcher. The card was then mounted on the Pi.[9]

The Pi was later attached to a TV display via HDMI cable and controlled with a keyboard and mouse. Wired keyboards caused the Pi to heat up significantly, so they were switched to wireless inputs.

Since we aimed to code our system in Python, we had to first install Python on the Pi. We logged in as a superuser by typing “sudo su” into the Pi’s terminal, and made sure our Pi was up-to-date by entering “apt-get update”. The virtual environment and PIP were installed by using the “apt-get install” command.[10,11]

Setting Up The Ultrasonic Sensor

The circuitry for the sensor required a basic set of jumper wires for a breadboard and some resistors to build it, primarily 300 and 470 ohm resistors. Since a 470 ohm resistor was unavailable, it was substituted with two 220 ohm and three 10 ohm resistors. The HC-SR04 ultrasonic sensor with four pins was used, although the only main ones we would use would be the ECHO and TRIGGER pins. The TRIGGER pin emits the ultrasonic wave while the ECHO pin receives the wave and sends the data back to the Pi. The complete wiring can be seen in Fig. 1.[12]

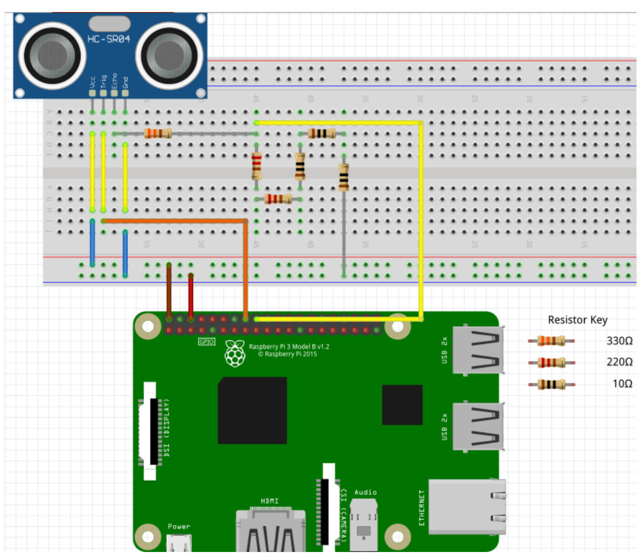

Fig. 1: Wiring for the ultrasonic sensor. The cluster of five resistors towards the right of the diagram was the replacement for the 470 ohm resistor we did not have.

The code for the ultrasonic sensor was the first priority in writing the code. Using Thonny IDE, the default Python Text Editor on Pi, we wrote our first code, depicted in Fig. 2. The “gpiozero” library was used as the syntax was more straightforward than others, such as “RPi.GPIO”.[10]

```
1 from gpiozero import DistanceSensor
2 import time
3
4 ultrasonic = DistanceSensor(echo=24,
      trigger=23)
5
6 while True:
7     print(ultrasonic.distance)
8     time.sleep(5)
```

Fig. 2. Code for testing the ultrasonic sensor. The first two lines import two libraries, “gpiozero” (allows Pi to use motors and sensors) and “time” (allows system to delay or pause the program). In line 4, the sensor is set up. This code will print the sensor’s reading to the console every five seconds. The sensor displays distances in meters by default.

Setting Up The Motor

A separate 9V battery had to be used in order to supply the motor with sufficient power. However the battery was added later in case an error was made while assembling the circuit. The circuit would need an L293D h-bridge so the motor could run in both directions. Essentially, power was given to the h-bridge, and the motor was connected to the h-bridge. The full circuit model can be seen in Fig. 3.[14]

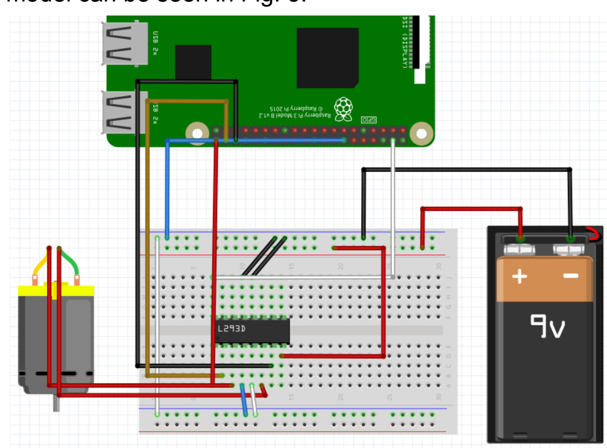

Fig. 3: Circuit diagram for the DC motor. This was done on a separate breadboard from the ultrasonic sensor.

The code was written after the motor was wired. Since the intruder would be closer to the sensor than the opposite wall, we figured the motor could release the PVC pipe when the ultrasonic sensor detected distance was less than the distance to the opposite wall. After releasing the pipe, the system would wait five seconds (as shown in Fig. 4 Line 11) before resetting the device (bringing the PVC pipe back up).[15]

```
1  from gpiozero import DistanceSensor
2  from gpiozero import Motor
3  import time
4  ultrasonic = DistanceSensor(echo=24, trigger=23)
5  motor = Motor(4, 14)
6  while True:
7      print("Triggered")
8      for i in range(1):
9          Motor.forward(1)
10         time.sleep(5)
11         Motor.backward(1)
12         time.sleep(5)
13     Motor.stop
```

Fig. 4. Draft code for running the motor. It will print "triggered" to the console, run the motor forward for five seconds, then backward for five seconds, then stop.

RESULTS

The finished system was able to sense an object moving in front of it within a range and return an accurate "True" signal to the Raspberry Pi which in turn gave signal to the motor that makes it move forward and backward. However, the motors used were not powerful enough to control the pvc and caused it to freefall.

DISCUSSION

This project helped us learn about how Python and Raspberry Pi could be used to control devices. Learning some basic engineering was also another notable benefit. In a further recreation of this project, a measure to ensure that the final product worked as a reliable security feature, a stronger motor should be used. Security everywhere always has the potential to be improved, as there is always going to be someone or something that can go around it.

AUTHOR INFORMATION

Present Addresses Aspiring Scholars Directed Research Program, 43505 Mission Blvd, Fremont, CA 94539

Author Contributions

All authors contributed to setting up the Pi's OS and display. Ashna Khemani wired and wrote the code needed for the ultrasonic sensor. Iniyan Joseph and Ashna Khemani wired the DC motor, and Joseph wrote the final code for it.

Notes This work was conducted over a two-month period.

ACKNOWLEDGMENT

We would like to acknowledge Olive Children's Foundation for sponsoring our project.

ABBREVIATIONS

GPIO: General Purpose Input/Output

REFERENCES

1) What is a Laser Sensor?. https://library.automationdirect.com/what-is-a-laser-sensor/(accessed Aug 16, 2019).
2) Little Laser Trip Wire Circuit. https://www.electroschematics.com/12362/little-laser-trip-wire-circuit/ (accessed Aug 16, 2019).
3) What is an ultrasonic sensor. https://www.keyence.com/ss/products/sensor/sensorbasics/ultrasonic/info/index.jsp (accessed Aug 16, 2019).
4) Are Bats Blind? https://askabiologist.asu.edu/echolocation/ (accessed Aug 15, 2019).
5) Burnett, R. Understanding How Ultrasonic Sensors Work. https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm (accessed Aug 16, 2019).
6) GPIO. https://www.renesas.com/kr/en/support/technical-resources/engineer-school/mcu-programming-peripherals-01-gpio.html (accessed Aug 15, 2019).
7) Admin. Raspberry Pi Technology, Working and Its Applications. https://www.watelectronics.com/know-all-about-raspberry-pi-board-technology/ (accessed Aug 16, 2019).
8) Download Raspbian OS. https://www.raspberrypi.org/downloads/raspbian/(accessed Aug 16, 2019).
9) How to install Raspbian on the Raspberry Pi. https://thepi.io/how-to-install-raspbian-on-the-raspberry-pi/ (accessed Aug 6, 2019).

10) iThoughts. How to install Python on Raspberry Pi. https://www.youtube.com/watch?v=VFs1dhJPYa4 (accessed Aug 7, 2019).

11) Raspberry Pi OS Installation Guide https://www.raspberrypi.org/documentation/installation/installing-images/README.md(accessed Aug 16, 2019).

12) Using a Raspberry Pi distance sensor (ultrasonic sensor HC-SR04). https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/ (accessed Aug 6, 2019).

13) Physical Computing with Python - Using an ultrasonic distance sensor. https://projects.raspberrypi.org/en/projects/physical-computing/13 (accessed Aug 8, 2019).

14) Controlling DC Motors Using Python With a Raspberry Pi. https://business.tutsplus.com/tutorials/controlling-dc-motors-using-python-with-a-raspberry-pi--cms-20051 (accessed Aug 16, 2019).

15) Basic Recipes. https://gpiozero.readthedocs.io/en/stable/recipes.html#motors (accessed Aug 16, 2019).