

# Desktop Application based Oil Tank Security System

Tithi Pal  
Dept. of ECE, JISCE

Trisit Ghosh  
Dept. of ECE, JISCE

Anasuya Sarkar  
Dept. ECE, JISCE

Sruti Basu  
Dept. of ECE, JISCE

Koyeli Majumder  
Dept. of ECE, JISCE

Sunita Pramanik  
Dept. of ECE, JISCE

Arindam Banerjee  
Dept. of ECE, JISCE

**Abstract**— In this paper, a desktop application has been developed to enable or disable security protection around oil tanks to overcome theft. From the desktop application, we can make the protection enabled or disabled. When the protection is enabled then if any person or animal goes near the oil tank then automatically the buzzer alarm starts blowing. Then the officers in charge can take immediate action against it. When the protection is disabled by the application then there is no alert sound if anyone approaches towards the tank. Ultrasonic sensor has been used for measuring the distance of the person or the animal. If the distance is less than a threshold value then only the alert buzzer is activated. The application is password protected and the administrator only can access the protection control. The application is interfaced with the microcontroller which is interfaced with the sensor and the alarm. The desktop application has been developed by Python language.

**Keywords**— Desktop Application, Ultrasonic Sensor, Microcontroller, Alarm Buzzer, Python

## I. INTRODUCTION

In the age of technological advancement, theft has been alarmingly increased. With the price hike of the fuels like petrol, diesel, other solvent oils, the oil theft has been increased. Due to pollution, green house effect and other atmospheric evolution, the underneath layer of the oil in our mother earth is getting down. In spite of that, the theft of oil has been increased a lot day by day and the oil is black marketed. That is why necessity of the oil preservation is a must. There are several literatures [1-3] on the oil security system design and all the systems incorporate security locally that means at the place of the incidence but there is no information or alert process to the central office from where the oil tanks are monitored.

Our work is purely a combination of hardware and software where the security system is controlled by an application. If any unknown person or animal approaches to the vicinity of the oil tank then automatically central office gets the alert information. But if some office staffs approaches toward the oil tank then he or she can disable the security system from the central office earlier and then can approach towards the oil tank. This idea is new to the best of our knowledge and

belief.

Here we aim to to enable or disable the protection around oil tanks to get rid of oil theft. In this equipment a password protected application is designed to enable or disable the security system. The password is only known to the main administrator. The security system is hardware based which is controlled by the application. The work is totally new to the best of the knowledge of the authors and therefore can't be compared with other similar works. The major limitation of the work is the delay in data transmission via cable for a large area. That is why our future planning is to design a mobile and a web application to control and monitor the system wirelessly.

The work has been organized as follows. Section II describes the desktop application. Section III describes the hardware application that is the interfacing of the application with the hardware. Section IV is the conclusion.

## II. DESKTOP APPLICATION

The application has been developed in Python. The flowchart of the proposed design is shown in Figure 1.

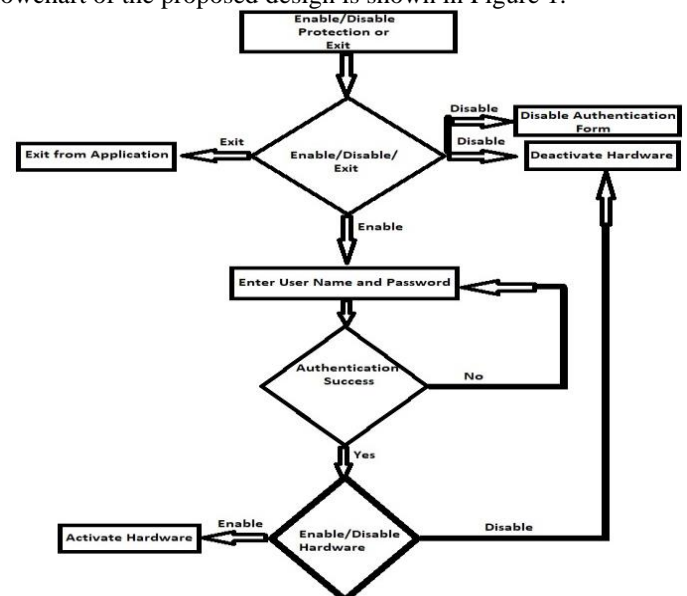


Figure 1 Flowchart of the proposed design

In this design, the application activates or deactivates the protection process from the main administrative office. The application is not accessible by all. Only the main administrator has the user name and password to enable or disable the protection which is shown in the flowchart in Figure 1. When the user wants to enable the protection then he or she must select "Enable" in the file menu and then log in to the system using the specified id and password given to the main administrator. If the person chooses "Disable" in the file menu, then the user authentication form is deactivated. If the user authentication is a success only then the protection can be enabled using the radio button switch given in the application. Initially the radio buttons are disabled. After successful authentication, the radio buttons are enabled. Then only the hardware system can be enabled. Here we have used a well known library "tkinter" to create the user application. To create the application, a window is created with the following codes "`tk=Tk()`", "`tk.title('Fuel Secutity System App')`" and "`tk.geometry('450x200')`". The first code is to create the class, second code is to create window title and the third code is to assign the size. "`tk.mainloop()`" maps the application interface in the main loop. Figures 2 to 8 show the user interface window for the application. There is a menu called "File" where 3 submenus are created "Enable", "Disable" and "Exit". When "enable" is selected then the "Log in" form is activated. Only administrator has the permission to get the user name and password. There is a button named "Authenticate" which is pressed to authenticate the user details. If the authentication is a success then only the protection control is enabled. User can forcefully disable the protection control by selecting "Disable" submenu. Also the user can exit from the application by selecting "Exit" submenu. To create menu and submenu, the following codes can be written.

```
menubar=Menu(tk)
file=Menu(menubar)
menubar.add_cascade(label="File",menu=file)
file.add_command(label="Enable",command=enable)
file.add_command(label="Disable",command=disable)
file.add_command(label="Exit",command=exitapp)
tk.config(menu=menubar)
```

"add\_cascade()" method creates menu items in the application. The second code creates submenu class inherited from menu class within one single menu item. "add\_command()" method creates submenu items and inside each submenu, a method is called which is triggered when the submenu is triggered. Finally the last code maps the menu item along with the submenus within the application. Any control (widget) can be disabled or enabled with the help of the following code.

```
widget.config(state=DISABLED) or
widget.config(state=NORMAL)
```

The authentication technique is totally restricted to the administrator. Only the administrator knows the username and the password to log in. There are two radio buttons ("Yes" and "No") which controls the protection in the hardware.

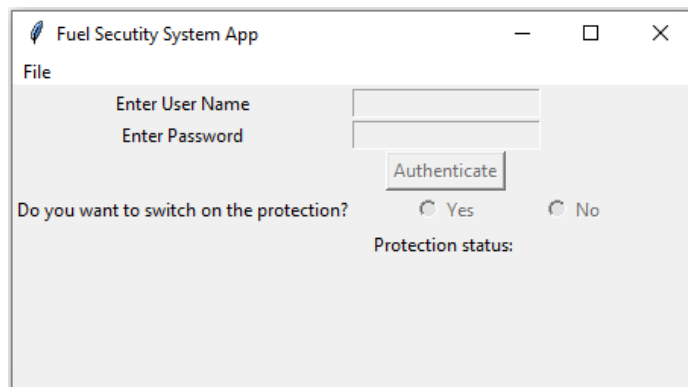


Figure 2 Application window for weather monitoring

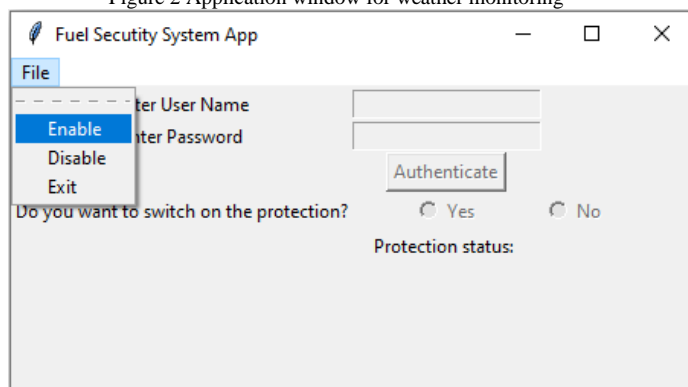


Figure 3 Menu for Logging in the application to control protection

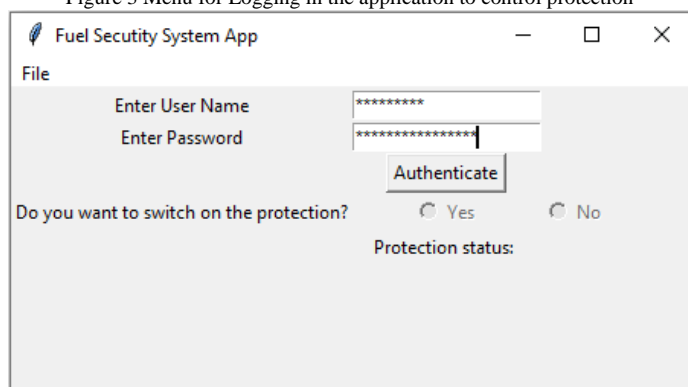


Figure 4 Log in process to control protection

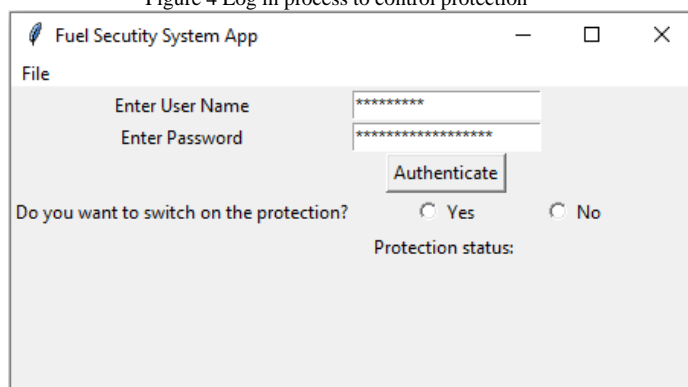


Figure 5 Protection activation after successful logging in

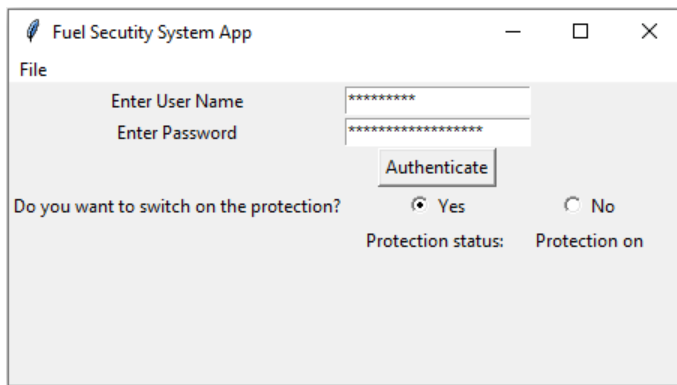


Figure 6 Protection Enabling/ Disabling Process

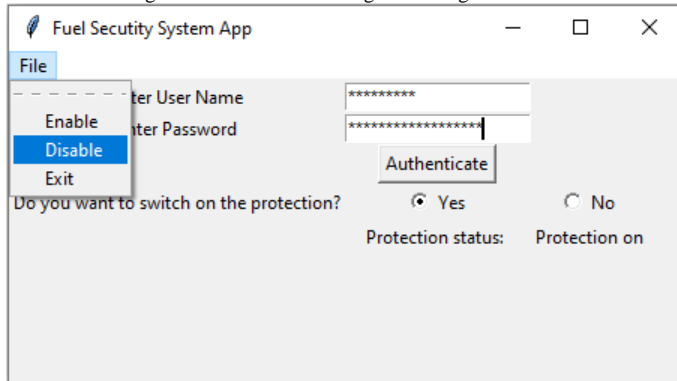


Figure 7 Protection Disabling Process

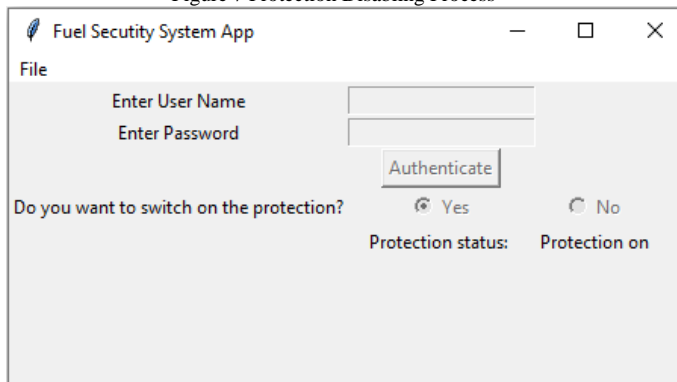


Figure 8 Application is disabled with the selection of Disable submenu. When any of the radio buttons is clicked, then the corresponding method inside it is triggered. Inside the methods, hardware interfacing codes are given. To interface the application with the hardware “pyserial” library has been used. In the “serial” class, port number and the baud rate is given. The following code is to identify the hardware “ser=serial.Serial(“COMX”,baudrate)”. “ser.write(stringdata.encode())” converts the Unicode string to byte and transfers to the microcontroller serial port.

### III. HARDWARE APPLICATION USING MICROCONTROLLER

To get the information for the presence of any living object, ultrasonic sensor based system is used here. The sensor is on or off based on application. When the sensor is on then if any object approaches towards the oil tank then the sensor measures the distance of the object and accordingly sends alert signal to the main administrative office (only when the distance is less than a specified value). But when the sensor is off then there is no opportunity to send alert. This part is purely controlled by the desktop application.

### A. Microcontroller Programming Fundamentals

In this work, as a microcontroller Arduino Uno AtMega328P has been used though Arduino Mega 2560 can also be used. Figure 9 shows the diagram for the microcontroller board. It is a 16MHz microcontroller and run by 5V volt DC power supply. The programming language used here is embedded C. For serial communication, the code “Serial.begin(9600)” is used. Here 9600 is the baud rate or the data transfer rate. “Serial.readString()” converts the byte code to Unicode string. This string is then converted to integer (if there is only integer in the string) with the help of “toInt()” method which is inbuilt in the “String” class. To detect intrusion, ultrasonic sensors have been used around the oil tank. There are four ultrasonic sensors used in the four directions (north, south, east and west) of the oil tank. The switching on and off conditions of the sensors are controlled by the applications (the power connection is controlled here). Moreover there is a alarm buzzer which is controlled by the ultrasonic sensor.

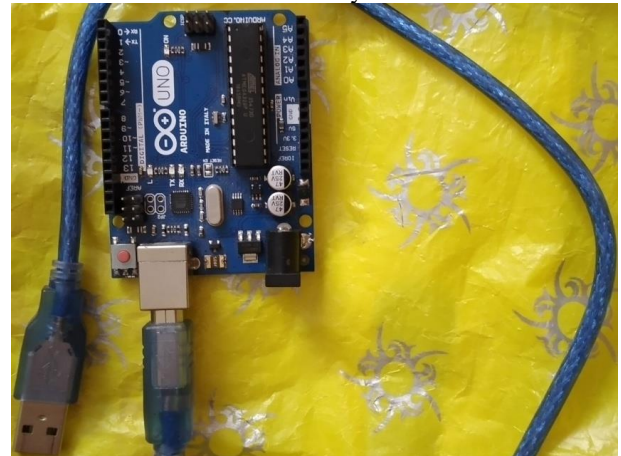


Figure 9 Arduino Uno AtMega328P microcontroller board

### B. Ultrasonic Sensor (HC-SR-04)

The ultrasonic sensor uses sound wave with ultrasonic range of frequency (almost in Mega Hertz range) which is inaudible. Figure 10 shows the ultrasonic sensor interfaced with the microcontroller. The sensor has the operating frequency 40 KHz. It has four terminals (power, ground, trigger and echo). From trigger terminal, a 12 microsecond periodic pulse with 10 microsecond pulse width is generated. This periodic pulse is reflected by the thief or any other animal (say obstacle) approaching towards it. The reflected wave is received by the echo pin. The distance (d) of the sensor from the obstacle can be measured with the following equation.

$$d = \frac{T \times \text{Velocity of sound}}{2} \quad (1)$$

Where, T=time taken by the wave between transmission and reception (measured in microsecond).

Here velocity of sound = 343 m/second=34300 cm/10<sup>6</sup> microsecond= 0.034 cm/microsecond. Thus,

$$d = \frac{T \times 0.034}{2} = T \times 0.017 \quad (2)$$





Figure 10 Ultrasonic sensor interfaced with Arduino microcontroller

### C. Overall Operation

At first we perform the operation to enable the security system. To do so we perform the software setting as shown in Figures 3 to 6. Then the protection is activated. There is a LED connected in place of alarm buzzer here. When the obstacle (here the hand) is less than or equal to 20 c.m. distant from the sensor then the LED is switched on which is shown in Figure 11.

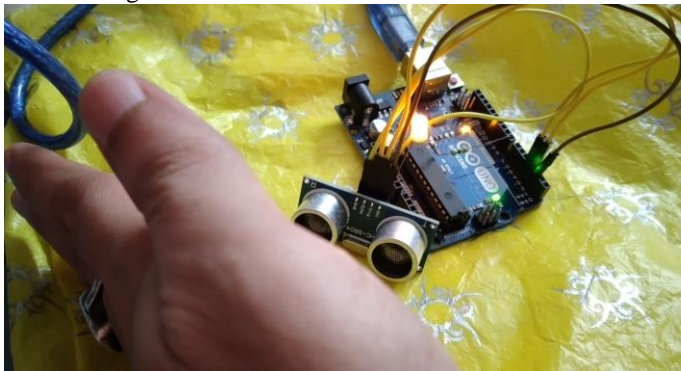


Figure 11 Protection is activated and sensor is sensing the obstacle

Next we perform the operation to disable the protection. To do so we follow the software setting as shown in Figures 6 and 7. As a result, the sensor is unable to sense the obstacle as shown in Figure 12.



Figure 12 Protection is deactivated and the sensor can't sense the obstacle  
The application has a third operation. If the user wants to exit from the application then there is a provision in the application. In the "File" menu, "Exit" submenu is to be pressed as shown in Figure 13. Then there is a message box display asking for the permission as shown in Figure 14. If "Yes" is pressed then the user can exit from the application as shown in Figure 15 (there is no application window, only the shell is displayed). This is achieved by "destroy()" method.

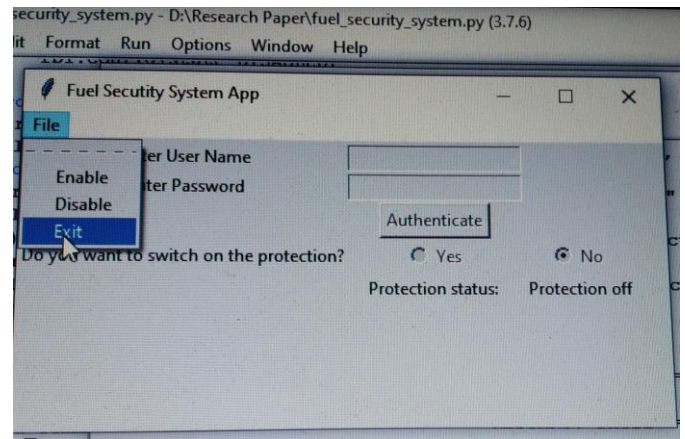


Figure 13 Exit operation in the application

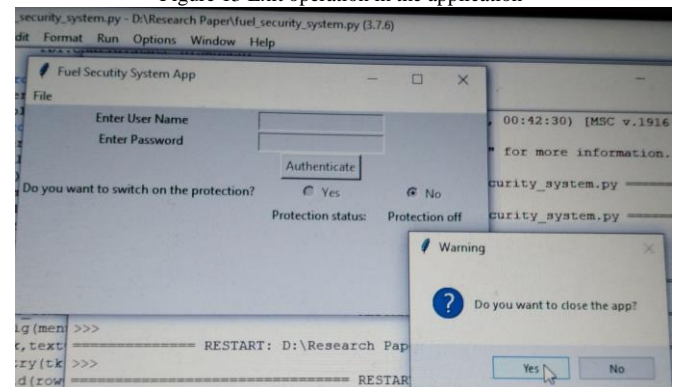


Figure 14 Message box display when the "Exit" submenu is selected

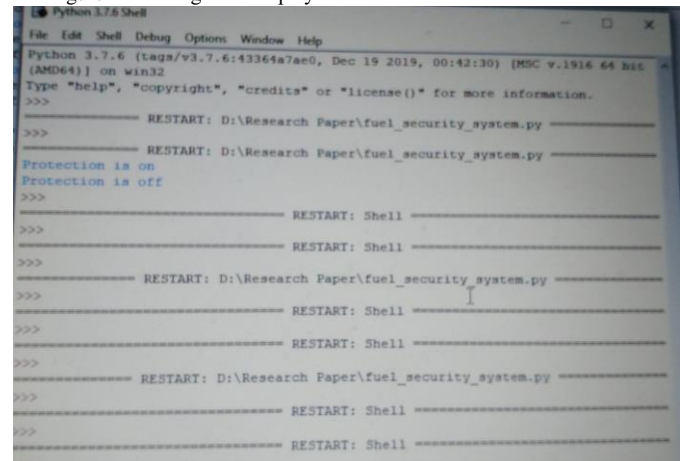


Figure 15 Application has been closed (destroyed)

In this paper, LED display is the indication of the presence of the obstacle but in place of LED, a buzzer or speaker interfaced with a tone generator is used. Figure 16 shows the tone generator circuit which is connected in place of the LED. The open pin of the circuit shown in Figure 16 is connected to the pin where the LED is actually connected.

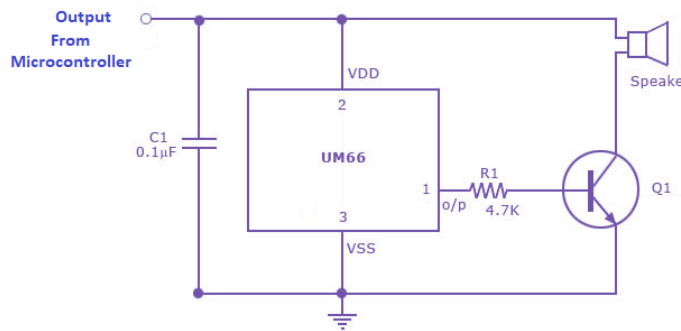


Figure 16 LED is replaced by the circuit to generate alert tone  
 UM66 is a tone generator IC. The IC has 64 note ROM memory. It is run by 1.5 to 4.5 V DC power supply. It is CMOS LSI based design. The tone generated is based on the package name shown at the body of the IC. When the output from the microcontroller is at logic high then both the IC and the transistor Q1 are biased. The transistor is of NPN type which means that it can be switched on by only logic high value. Now when the output of the IC is at logic high then the transistor Q1 is turned on and the speaker starts sound based on the tone frequency.

#### IV. CONCLUSION

In this paper, we designed a security system for oil tank to get rid of theft. The system had two parts, (i) a software application and a hardware application. The software application was a desktop application developed in python and it was for enabling or disabling the protection around the oil tank. In the hardware part, we interfaced the alert system with the microcontroller to act with the command coming from the software application. Though there were related similar works reported earlier but in this work the protection was administered from the office and also the alert goes to the office directly. In this context, the work was totally new to the best of the knowledge of the authors and so could not be compared with any other existing works.

#### ACKNOWLEDGMENT

The authors express their heartiest gratitude to the authorities of JIS College of Engineering for giving the opportunity to perform the research.

#### REFERENCES

- [1] Diesel Fuel Security Guide, <https://www.centretank.com/diesel-fuel-security-advice>.
- [2] Fuel Security System, <https://www.tebratrading.com/fuel-security-devices>.
- [3] Fuel Theft Alarm, <http://www.digitalsystems.lv/en/fuel-theft-alarm>.