

## Rosu Alexandru



**Title:** Device for measuring the water level from a recipient

**Team:** Rosu Alexandru

Popa Paul(1305A),

Razvan Sarbu(1305A)

Eduard Gavriiliuc(1306B)

**Description:** We build an application that measures the water level from a recipient and sends an alert email.

### Hardware Components:

Raspberry Pi 3 - Model B;

Ultrasonic Sensor - HC-SR04 (Generic);

Sound Sensor

Buzzer

Jumper Wires

BreadBoard

### Software:

Raspbian OS

Python

**How it works:** Firstly we have implemented a sound sensor. If the sensor is receiving any sounds, the ultrasonic sensor starts working, it measures the distance between sensor and water. If the distance is less than 4cm , then we send an email and the buzzer is activated.

Sound.py

```
import distanta
```

```
import RPi.GPIO as GPIO
```

```
import time
```

```
channel = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(channel, GPIO.IN)

def callback(channel):
    if GPIO.input(channel):
        print "Detecteaza sunet!"
distanta.distanta()
GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime = 300)
GPIO.add_event_callback(channel, callback)

while True:
    time.sleep(1)
```

Distance.py

```
import mail
import RPi.GPIO as GPIO
import time

def distanta():
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)

    TRIG = 2
    ECHO = 3
    i=0

    GPIO.setup(TRIG ,GPIO.OUT)
    GPIO.setup(ECHO,GPIO.IN)
    GPIO.setup(4 ,GPIO.OUT)

    GPIO.output(TRIG, False)
    print("Starting.....")
    time.sleep(2)

    while True:
        GPIO.output(TRIG, True)
```

```

time.sleep(0.00001)
GPIO.output(TRIG, False)

while GPIO.input(ECHO)==0:
    pulse_start = time.time()

while GPIO.input(ECHO)==1:
    pulse_stop = time.time()

pulse_time = pulse_stop - pulse_start

distance = pulse_time * 17150
print(round(distance, 2));

time.sleep(1)

if distance < 4:
    print("Water will overflow")
mail.gmail(distance)
    GPIO.output(4, True);
    time.sleep(0.5)
    GPIO.output(4, False);
    time.sleep(0.5)
    GPIO.output(4, True);
    time.sleep(0.5)
    GPIO.output(4, False);
    time.sleep(0.5)
else:
    GPIO.output(4, False);

```

### Steps for Sending Email using Raspberry Pi

**Step 1:-** Setting up the raspberry pi module- connect the power cable and LAN cable to raspberry pi then create WIFI hotspot and connect with it.

**Step 2:-** After then open the terminal window on Pi. Then, open the putty software and paste the host name or ip address.

**Step 3:-** SMTP configuration

```
# Config file for sSMTP sendmail
```

```
#
```

```
# The person who gets all mail for userids < 1000
```

# Make this empty to disable rewriting.

root=postmaster

# The place where the mail goes. The actual machine name is required no

# MX records are consulted. Commonly mailhosts are named mail.domain.com

mailhub=smtp.mail.com:465

# Where will the mail seem to come from?

#rewriteDomain=

# The full hostname

hostname=raspberrypi

AuthUser = mishulean20@gmail.com

AuthPass = Mishu2020

UseSTARTTLS=YES

# Are users allowed to set their own From: address?

# YES - Allow the user to specify their own From: address

# NO - Use the system generated From: address

#FromLineOverride=YES

Mail.py

import smtplib

def gmail(nivel):

server=smtplib.SMTP('smtp.gmail.com',587)

server.starttls()

server.login("mishulean20@gmail.com", "Mishu2020")

msg="merge "+str(nivel)

server.sendmail("mishulean20@gmail.com","razvansarbu555@gmail.com",msg)

server.quit()