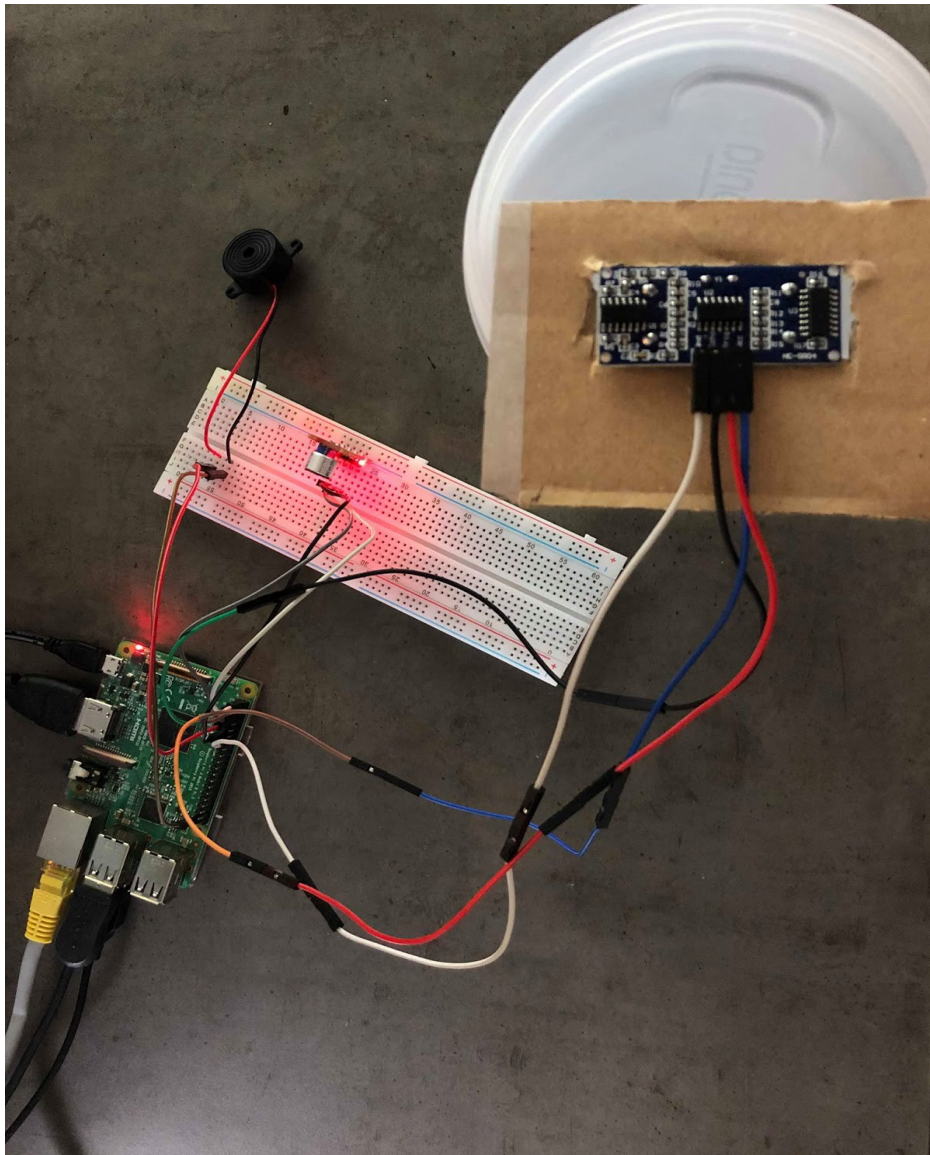


## Device for measuring the water level from a recipient



### **Team:**

Rosu Alexandru(1309B),  
Popa Paul,  
Razvan Sarbu,  
Eduard Gavriiuc(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

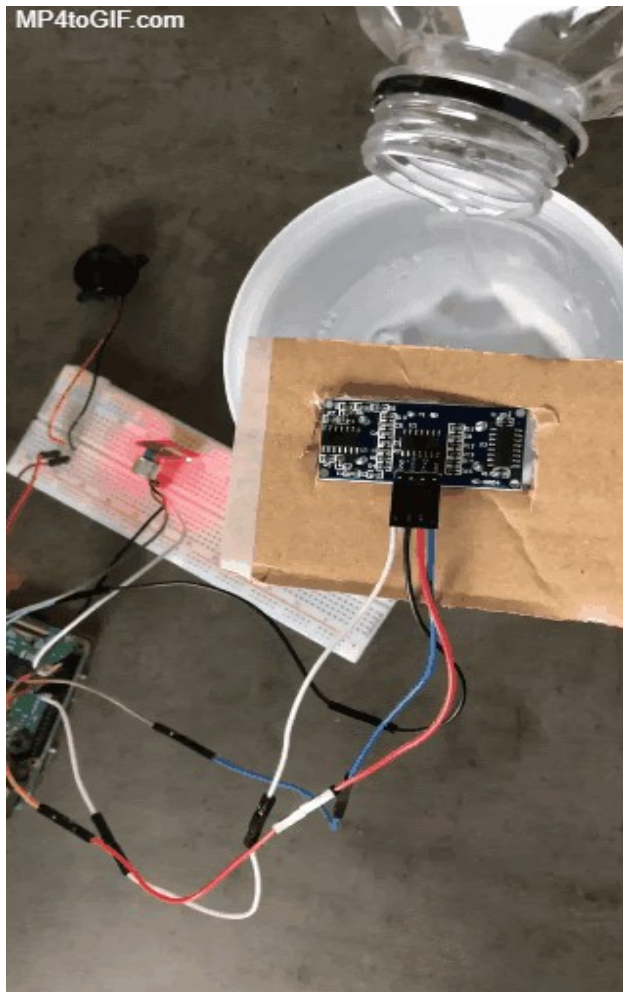
**Software:**

Raspbian OS

Python

**How it works:**

Firstly, we have connected the sensors to a Raspberry Pi 3 board using jumper wires and an additional breadboard as following: a 3-pin sound sensor; if it's receiving any sounds, the next sensor that we have used starts working, the HC-S04 ultrasonic sensor, which measures the distance between sensor and water; if the water level is increasing and the distance is less than 4cm leads us to the last thing that we've implemented: a simple actuator, a buzzer and an alert mail is send with the exact value of the distance.



Short Demo

```
pi@raspberrypi: ~/proiectSM
File Edit Tabs Help
pi@raspberrypi:~/proiectSM $ ls
afisare.py afisare.pyc distanta.py distanta.pyc mail.py mail.pyc
pi@raspberrypi:~/proiectSM $ python sound.py
Detecteaza sunet!
Starting.....
14.24
14.68
14.63
14.58
14.22
14.25
14.66
14.65
7.65
6.89
5.87
5.95
5.92
5.56
5.69
4.92
4.41
4.35
3.77
Water will overflow
```

## Sound.py

```
pi@raspberrypi: ~/proiectSM
File Edit Tabs Help
GNU nano 3.2 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)
```

## Distanta.py

```
pi@raspberrypi: ~/proiectSM
File Edit Tabs Help
GNU nano 3.2          distante.py

import mail
import RPi.GPIO as GPIO
import time

def distante():
    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);
```

## Mail.py

```
>_ pi@raspberrypi: ~/pr... [pi]
pi@raspberrypi: ~/proiectSM
File Edit Tabs Help
GNU nano 3.2          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()
```

Configurare SSMTP:

pi@raspberrypi: ~/proiectSM

File Edit Tabs Help

GNU nano 3.2

/etc/ssmtp/ssmtp.conf

```
#
# 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
```