

Echipa:

Zalinca Claudiu-Serban, Gurzun Andrei, Condrea Octavian-Ioan, Balan Marius -

## ‘Home smart alarm’

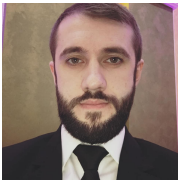
### Zalinca Claudiu-Serban



*Sarcini:* configurare hardware si implementare generala a codului

*Email:*

### Gurzun Andrei



*Sarcini:* documentatie si desen hard

*Email:* [andrew\\_3z@yahoo.com](mailto:andrew_3z@yahoo.com)

### Condrea Octavian-Ioan



*Sarcini:* implementare SiriControl

*Email:* [octav9999x@gmail.com](mailto:octav9999x@gmail.com)

### Balan Marius



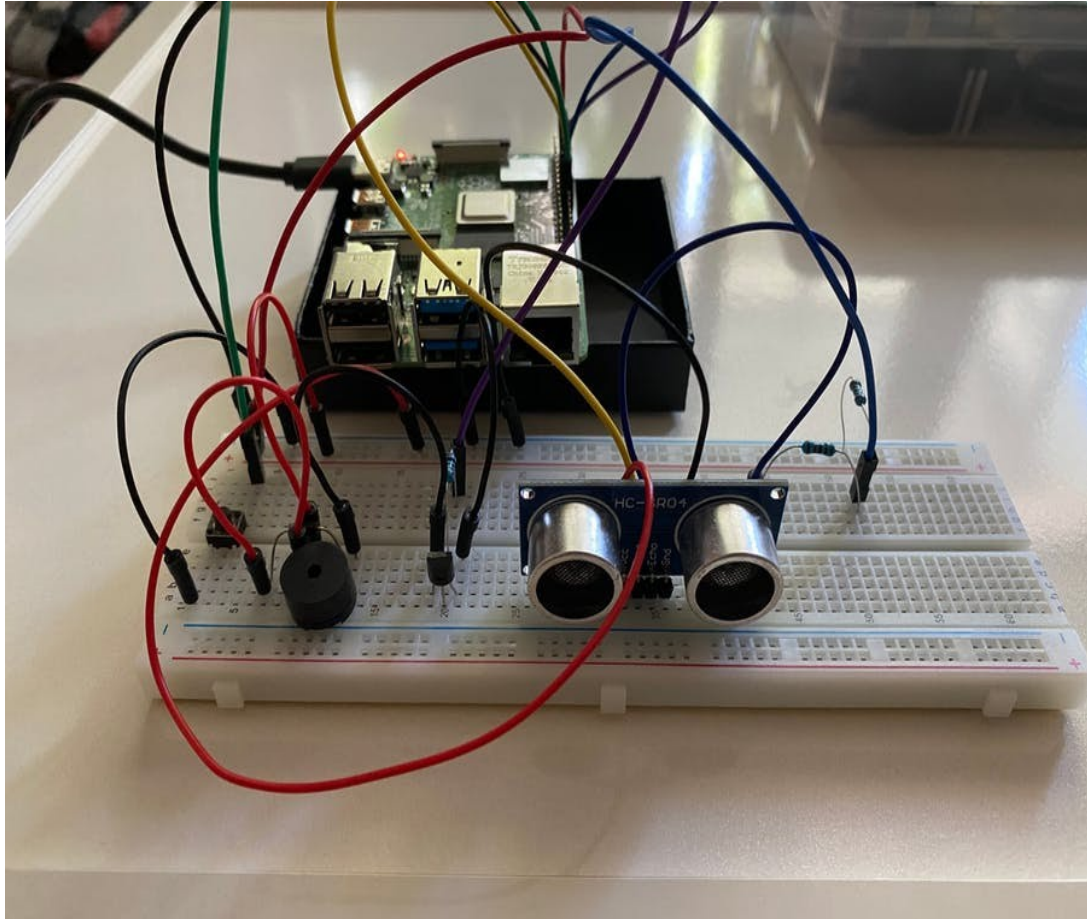
*Sarcini:* notificare email via SMTP + senzor distanta -> calcul distanta

*Email:* [ballan.marius@yahoo.com](mailto:ballan.marius@yahoo.com)

## Titlu proiect: 'Home smart alarm'

<https://www.hackster.io/340038/home-smart-alarm-2b0568>

- Name: Home smart alarm
- Cover Image:



- Story:

We set out to create an alarm system for motion detection. Not only is it extremely practical but it is also very useful for anyone who wants to create a safe and secure space for their family. This project has not only a single use, but promises to make the lives of several people easier (e.g. a blind person can be warned when an obstacle arises in front of him).

## What does the project do:

- Detects motion
- Starts an alarm in case of detection
- Email notification
- You can shut down the alarm by using Voice Recognition or by pressing the button

## Steps to do the project:

- Install pigpio

First, if a previous **pigpio.zip** file exists in the current directory, then delete it:

```
rm pigpio.zip
```

Then delete if a previous **PIGPIO** directory exists **in the pwd**:

```
sudo rm -rf PIGPIO
```

We then download the latest version of the **pigpio.zip** archive with the **wget** command (web get):

```
wget abyz.me.uk/rpi/pigpio/pigpio.zip
```

Unzip the file:

```
unzip pigpio.zip
```

Change the directory.

```
cd PIGPIO
```

Then install it with the following commands:

```
make  
sudo make install
```

We need to start it using:

```
sudo pigpiod
```

- Install SiriControl-System

-> git clone <https://github.com/thesanjeetc/SiriControl-System>

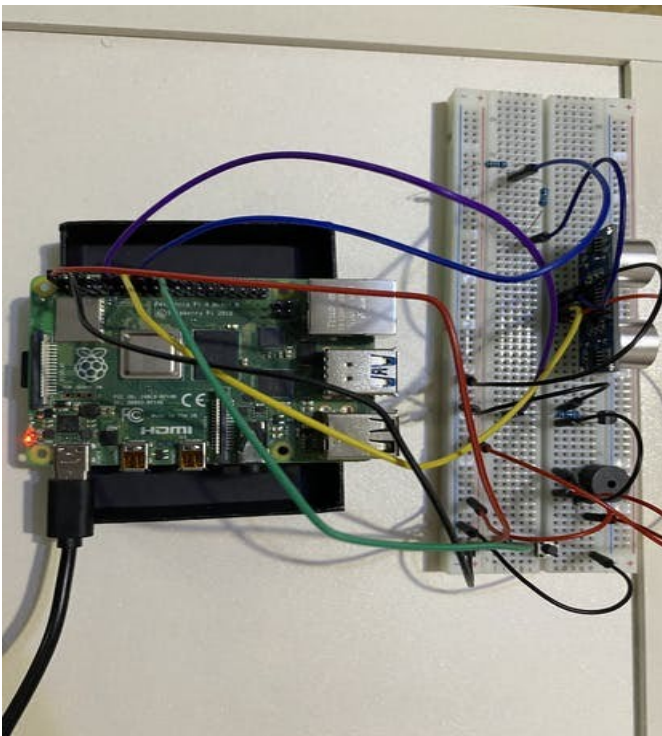
- Email Configuration for Voice Control

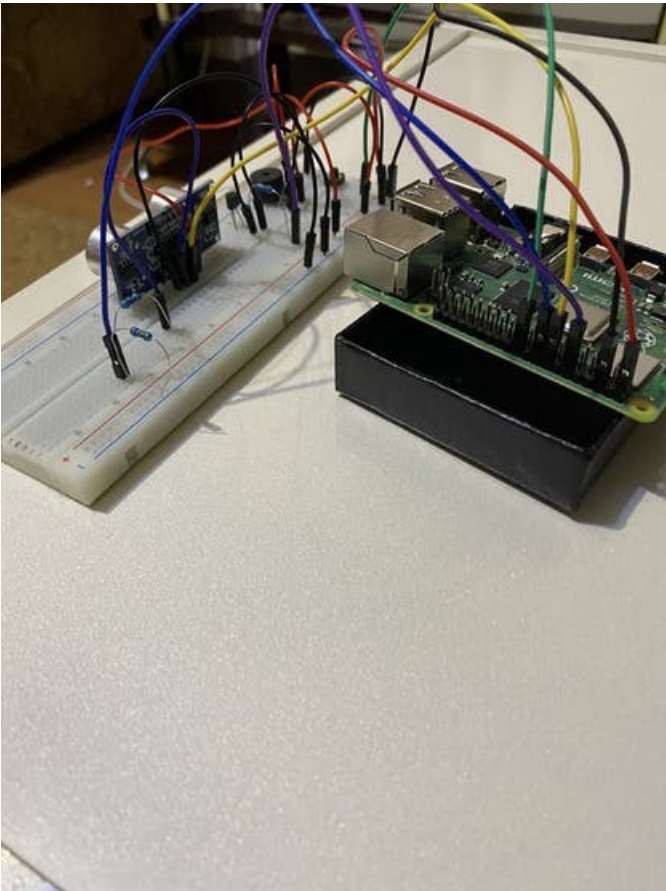
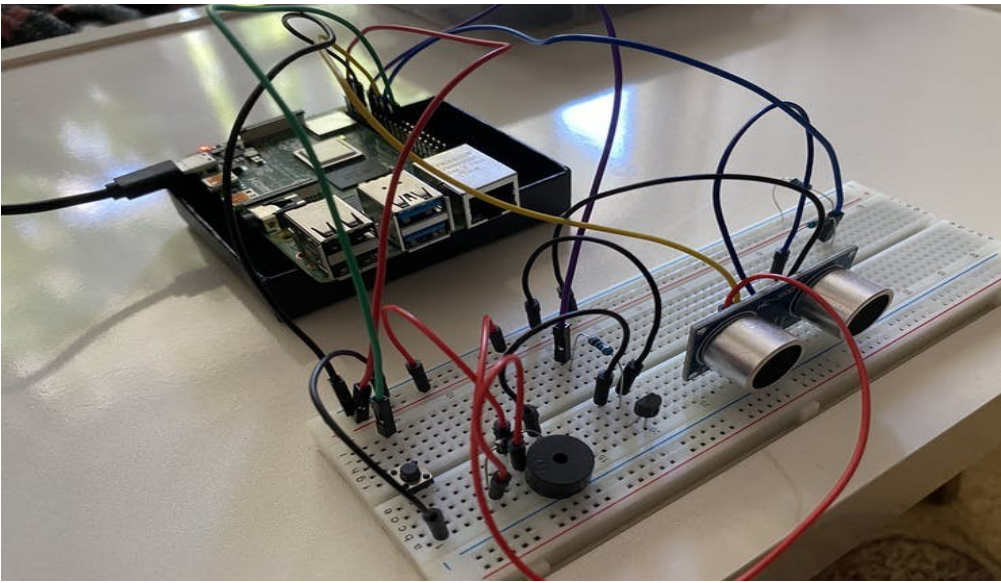
-> <https://www.instructables.com/id/SiriControl-Add-Siri-Voice-Control-to-Any-Raspberr/>

Add a new module in the modules folder from SiriControl-System (code in Attachments)

- Fill both the email and password fields in the mainfile and also in the SiriControl-System
- Copy-paste the main code in a file with extension.py
- Run the code: `python3 nameOfTheFile.py`
- Run the SiriControl-System: `python siricontrol.py` (SiriControl-System folder)

**Hardware Configuration:**





## How does the project work:

- Detects Motion

<https://youtu.be/iSdZGEBc9aY>

<https://youtu.be/nvLO1nseqng>

- Email notification

<https://youtu.be/hI2WXHKFclc>

- Voice Control

[https://youtu.be/l\\_kuX5jFXcM](https://youtu.be/l_kuX5jFXcM)

## Hardware components

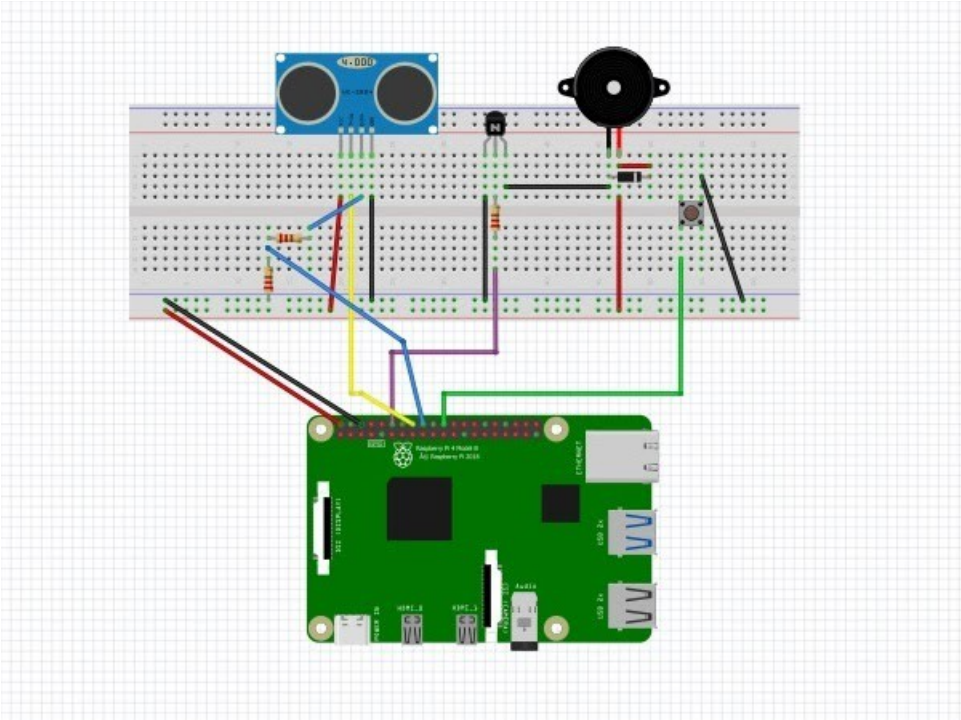
- Raspberry Pi 4 Model B
- Male/Female Jumper Wires
- Male/Male Jumper Wires
- Ultrasonic Sensor - HC-SR04
- Buzzer
- Pushbutton
- Resistors 1k, 2k
- Diode 1N4007

## Software apps and online services

- SiriControl: <https://github.com/thesanjeetc/SiriControl-System>

SiriControl is an open source framework designed with developers in mind. It provides a simple way of using Siri commands by dynamically loading modules created by the user.

**Schematics:**



## Code

```
import time
import RPi.GPIO as GPIO
import pigpio
import smtplib
import ssl

port=465
smtp_server = "smtp.gmail.com"
sender_email= ""
receiver_email= ""
password= ""
message = ""\
    Subject: ALARM!!!

    Motion Detected in your house.""

context=ssl.create_default_context()

pi=pigpio.pi()
GPIO.setmode(GPIO.BOARD)
trig=16
echo=18
buzzer=18
button=25
GPIO.setup(trig,GPIO.OUT)
GPIO.setup(echo,GPIO.IN)
pi.set_mode(buzzer,pigpio.OUTPUT)
pi.set_pull_up_down(button,pigpio.PUD_UP)

def calculateDist():
    GPIO.output(trig,GPIO.LOW)
    time.sleep(0.05)
```



```

GPIO.output(trig,GPIO.HIGH)
time.sleep(0.00001)
GPIO.output(trig,GPIO.LOW)

start=time.time()
stop=time.time()

while GPIO.input(echo)==0:
    start=time.time()

while GPIO.input(echo)==1:
    stop=time.time()

duration=stop-start

distance=34300/2*duration

#if distance < 3400:
    #print("Distance = %.2f" % distance)
return distance

print("The system calibrating...")
i=0
val=0
while i<20:
    val=val+calculateDist()
    i=i+1
    #print(i)
val=val/20
print("Average value: %.2f" % val)

try:
    while True:
        ti=0
        ok=True
        dist=calculateDist()
        eroare=val-dist

```

```

if(eroare<0):
    eroare=eroare*(-1)
while ok:
    #print("Everything is fine!")
    dist=calculateDist()
    eroare=val-dist
    if(eroare<0):
        eroare=eroare*(-1)
    if(eroare<5*val/100):
        ok=True
    else:
        if(ti==0):
            ok=True
            ti=ti+1
        else:
            ok=False
pi.hardware_PWM(buzzer,500,500000)
with smtplib.SMTP_SSL(smtp_server,port,context=context) as server:
    server.login(sender_email,password)
    server.sendmail(sender_email,receiver_email,message)
i=0
while pi.read(button)==1:
    if pi.read(buzzer)==0:
        start=time.time()
        while pi.read(buzzer)==0:
            stop=time.time()
            print(stop-start)
            i=i+1
            if(stop-start>1):
                break
        if(stop-start>1):
            break
        #print("ALARMA %d" % ti)
    pi.hardware_PWM(buzzer,0,0)
except KeyboardInterrupt:
    pass

```

```
pi.write(buzzer,0)
pi.stop()
GPIO.cleanup()
```

## Module in SiriControl

```
#You can import any required modules here
import RPi.GPIO as GPIO
#This can be anything you want
moduleName = "AlarmOff"

#All of the words must be heard in order for this module to be executed
commandWords = ["turn","off","alarm"]

def execute(command):
    #Write anything you want to be executed when the commandWords are heard
    #The 'command' parameter is the command you speak
    buzzer=12

    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(buzzer,GPIO.OUT)
    GPIO.output(buzzer,GPIO.LOW)

    return
```