Sending data from ESP to email account

Problem Statement: Connect sensor to ESP32. Monitor the sensor value, if the sensor value increased above threshold, then send the email to the user

~By Uday, Shweta

LinkedIn

https://www.linkedin.com/in/uday-yeshi-79240127a//

https://www.linkedin.com/in/shweta-shinde-018b8b27a//

Mail

uday.yeshi23@spit.ac.in

shweta.shinde23@spit.ac.in

SMTP

- **SMTP** means *Simple Mail Transfer Protocol* and it is an internet standard for email transmission. To send emails using an ESP32, you need to connect it to an SMTP Server

uMail Module

- To easily send emails with MicroPython, we'll use a module called uMail. This module is not part of the standard collection of MicroPython libraries, so we'll need to upload it separately to our board.

SMTP Server Settings

Gmail SMTP Server Settings

If you're using a Gmail account, these are the SMTP Server details:

- SMTP Server: **smtp.gmail.com**
- SMTP username: Complete Gmail address
- SMTP password: Your Gmail password
- SMTP port (TLS): **587**
- SMTP port (SSL): **465**
- SMTP TLS/SSL required: **yes**

Live or Hotmail SMTP Server Settings

For Live or Hotmail accounts, these are the SMTP Server settings:

- SMTP Server: **smtp.live.com**
- SMTP Username: Complete Live/Hotmail email address
- SMTP Password: Your Windows Live Hotmail password
- SMTP Port: **587**
- SMTP TLS/SSL Required: **Yes**

Outlook SMTP Server Settings

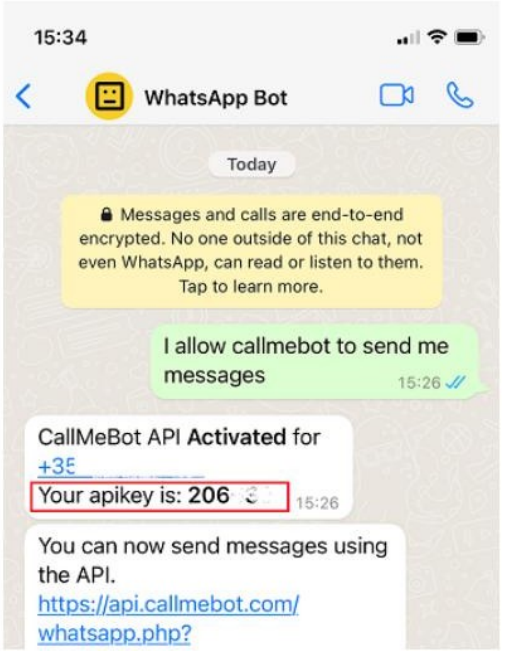
For Outlook accounts, these are the SMTP Server settings:

- SMTP Server: **smtp.office365.com**
- SMTP Username: Complete Outlook email address
- SMTP Password: Your Outlook password
- SMTP Port: **587**
- SMTP TLS/SSL Required: **Yes**

Using WhatsApp

CallMeBot WhatsApp API

1. Add the phone number +34 644 51 95 23 to your Phone Contacts. (Name it as you wish);
2. Send the following message: "*I allow callmebot to send me messages*" to the new Contact created (using WhatsApp of course);
3. Wait until you receive the message "*API Activated for your phone number. Your APIKEY is XXXXXX*" from the bot.



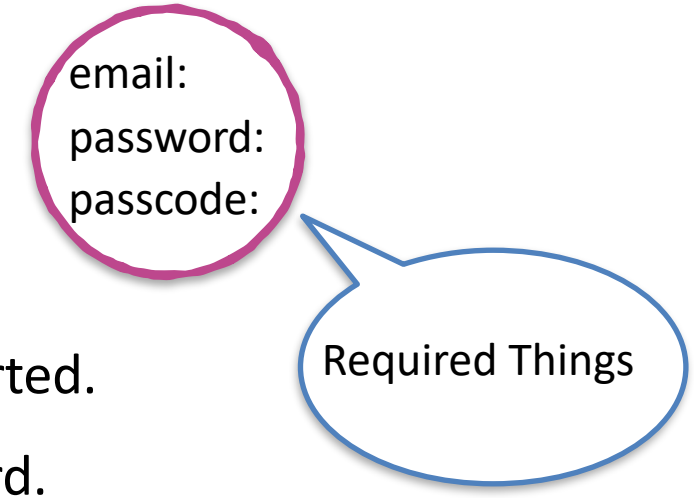
```
try:
  import urequests as requests
except:
  import requests

import network

import esp
esp.osdebug(None)

import gc
gc.collect()

#Your network credentials
ssid = 'REPLACE_WITH_YOUR_SSID'
password = 'REPLACE_WITH_YOUR_SSID'
```

```
phone_number = 'YOUR_PHONE_NUMER_INTERNATIONAL_FORMAT'
#Your callmebot API key
api_key = 'CALLMEBOT_API_KEY'

def connect_wifi(ssid, password):
  #Connect to your network
  station = network.WLAN(network.STA_IF)
  station.active(True)
  station.connect(ssid, password)
  while station.isconnected() == False:
    pass
  print('Connection successful')
  print(station.ifconfig())

def send_message(phone_number, api_key, message):
  #set your host URL
  url = 'https://api.callmebot.com/whatsapp.php?phone='+phone_numbe
```

```
#make the request
response = requests.get(url)
#check if it was successful
if response.status_code == 200:
  print('Success!')
else:
```

Step1: creating of Gmail account for giving access to ESP32

- Go to gmail page and create new account
- Open your Google Account.
- In the navigation panel, select Security.



- Under “Signing in to Google,” select 2-Step Verification > Get started.
- After enabling 2-step verification, you can create an app password.
- Open your Google Account.
- In the navigation panel, select Security.
- Under “Signing in to Google,” select App Passwords.
- In the Select app field, choose mail. For the device, select Other and give it a name, for example ESP32. Then, click on Generate. It will pop-up a window with a password that you’ll use with the ESP32 or ESP8266 to send emails. Save that password (even though it says you won’t need to remember it) because you’ll need it later.

umail.py

- Create a new file in your IDE with the name `umail.py` and paste code from moodle there. Save that file

Boot.py

```
import network, time, machine
ssid = ‘SSID'
password = ‘password'

station = network.WLAN(network.STA_IF)

station.active(True)
station.connect(ssid, password)

while station.isconnected() == False:
  pass

print('Connection successful')
print(station.ifconfig())
```

Main.py

```
import umail
import network, time, machine
sender_email = ‘sender email'
sender_name = 'ESP32' #sender name
sender_app_password = ‘password'
recipient_email =‘recipient email'
email_subject ='Hello'

print("Hello")
# Send the email
smtp = umail.SMTP('smtp.gmail.com', 465, ssl=True) # Gmail's SSL port
smtp.login(sender_email, sender_app_password)
smtp.to(recipient_email)
smtp.write("From:" + sender_name + "<"+ sender_email+">\n")
smtp.write("Subject:" + email_subject + "\n")
smtp.write("Hello ESP32 " "\n")
smtp.send()
smtp.quit()
```

Status on Gmail account