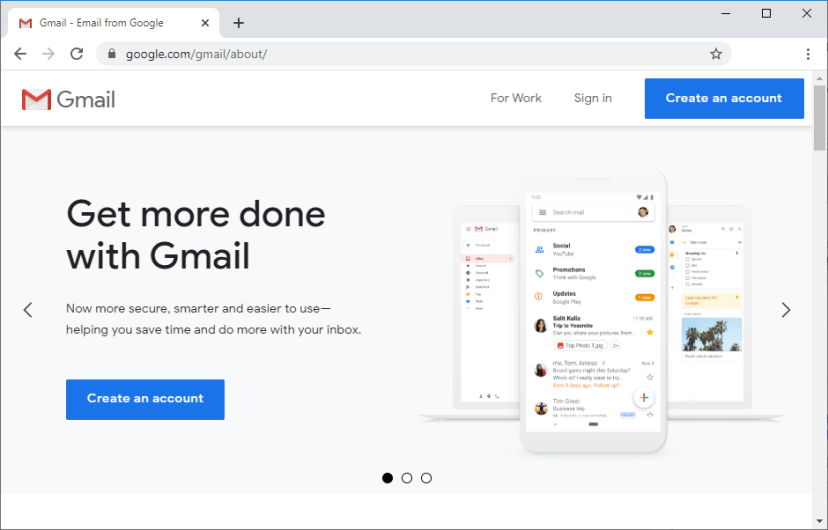
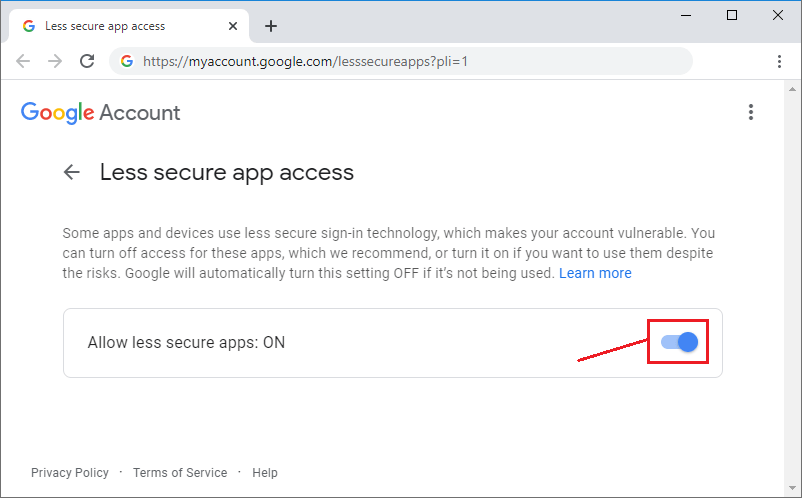
<https://randomnerdtutorials.com/esp32-cam-send-photos-email/>



Allow less secure apps to get access to this new Gmail account, so that you’re able to send emails. You can [**open this link**](https://myaccount.google.com/lesssecureapps?pli=1) to go to that menu.



SEND TO GMAIL

==============

SMTP Server: smtp.gmail.com

SMTP username: gmail address

SMTP password: Your Gmail password

SMTP port (TLS): 587

SMTP port (SSL): 465

SMTP TLS/SSL required: yes

SEND TO OUTLOOK

===============

SMTP Server: smtp.office365.com

SMTP Username: Outlook email address

SMTP Password: Your Outlook password

SMTP Port: 587

SMTP TLS/SSL Required: Yes

Live or hotmail

===============

SMTP Server: smtp.live.com

SMTP Username: Live/Hotmail email address

SMTP Password: Your Windows Live Hotmail password

SMTP Port: 587

SMTP TLS/SSL Required: Yes

Include library ESP32 Mail Client

#include "ESP32\_MailClient.h"

#define emailSenderAccount "ACCOUNT@gmail.com"

#define emailSenderPassword "PASSWORD"

#define smtpServer "smtp.gmail.com"

#define smtpServerPort 465

#define emailSubject "ESP32-CAM Photo Captured"

#define emailRecipient "SendTo@gmail.com"

SMTPData smtpData;

void sendPhoto( void ) {

smtpData.setLogin(smtpServer, smtpServerPort, emailSenderAccount, emailSenderPassword);

// Set the sender name and Email

smtpData.setSender("ESP32-CAM", emailSenderAccount);

// Set Email priority or importance High, Normal, Low or 1 to 5 (1 is highest)

smtpData.setPriority("High");

// Set the subject

smtpData.setSubject(emailSubject);

// Set the email message in HTML format

smtpData.setMessage("<h2>Photo captured with ESP32-CAM and attached in this email.</h2>", true);

// Set the email message in text format

//smtpData.setMessage("Photo captured with ESP32-CAM and attached in this email.", false);

// Add recipients, can add more than one recipient

smtpData.addRecipient(emailRecipient);

//smtpData.addRecipient(emailRecipient2);

// Add attach files from SPIFFS

smtpData.addAttachFile(FILE\_PHOTO, "image/jpg");

// Set the storage type to attach files in your email (SPIFFS)

smtpData.setFileStorageType(MailClientStorageType::SPIFFS);

smtpData.setSendCallback(sendCallback);

if (!MailClient.sendMail(smtpData))

Serial.println("Error sending Email, " + MailClient.smtpErrorReason());

smtpData.empty();

}

// Photo File Name to save in SPIFFS

#define FILE\_PHOTO "/photo.jpg

capturePhotoSaveSpiffs();

sendPhoto();

// Capture Photo and Save it to SPIFFS

void capturePhotoSaveSpiffs( void ) {

camera\_fb\_t \* fb = NULL; // pointer

bool ok = 0; // if the picture has been taken correctly

do {

// Take a photo with the camera

fb = esp\_camera\_fb\_get();

if (!fb) {

Serial.println("Camera capture failed");

return;

}

// Photo file name

Serial.printf("Picture file name: %s\n", FILE\_PHOTO);

File file = SPIFFS.open(FILE\_PHOTO, FILE\_WRITE);

// Insert the data in the photo file

if (!file) {

Serial.println("Failed to open file in writing mode");

}

else {

file.write(fb->buf, fb->len); // payload (image), payload length

Serial.print("The picture has been saved in ");

Serial.print(FILE\_PHOTO);

Serial.print(" - Size: ");

Serial.print(file.size());

Serial.println(" bytes");

}

// Close the file

file.close();

esp\_camera\_fb\_return(fb);

// check if file has been correctly saved in SPIFFS

ok = checkPhoto(SPIFFS);

} while ( !ok );

}

// Check if photo capture was successful

bool checkPhoto( fs::FS &fs ) {

File f\_pic = fs.open( FILE\_PHOTO );

unsigned int pic\_sz = f\_pic.size();

return ( pic\_sz > 100 );

}