Post image to LINE with ESP32 [M5Stack-TimerCamera]

- HTTPS (TLS) client implementation
- Image posting using LINE-API

Table of contents

1. Equipment selection

2. Development environment

3. Arduino IDE settings

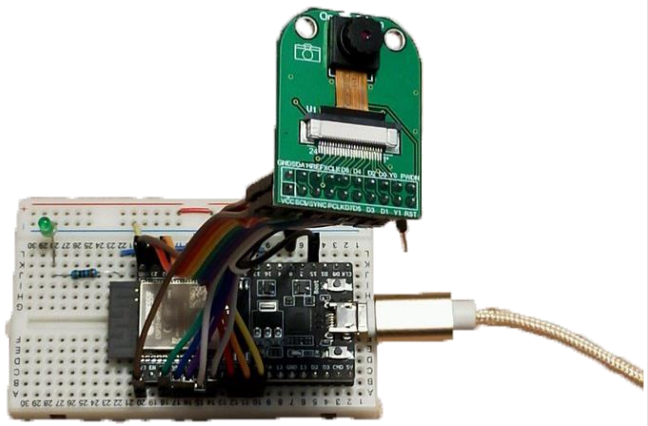
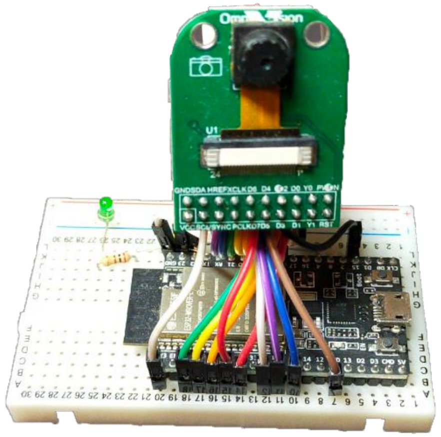
4. LINE settings (token acquisition)

5. Arduino program

6. Program writing, operation check

1-1. Equipment selection (target around 4000 yen or less)

*Please note that the cost varies depending on the season.

	Same hardware configuration		Similar hardware configuration	
	①ESP32 (WROOM) and OV2640	②M5Stack UnitCam (OV2640)	③ESP32 (WROVER) and OV2640	④M5Stack TimerCamera (OV3660)
Figure				
Specification	Memory [SRAM]: 520kbyte, Resolution: 2M pixel		Memory [SRAM]: 8Mbyte	
		プログラム書込にはキットが必要*1	Resolution: 2M pixel	Resolution: 3M pixel
Usage	Image		Image, Video	
Cost	¥3930	M5Stack: UnitCam $18.95 [marutsu: ¥2946] + ¥1100*1	¥4080	M5Stack: F)$19.95, X)$17.95 [SwitchSience: F)¥2860, X)¥2596]
Soft	Almost Usable (There are differences in Arduino motherboard settings and port usage)			
Youtube Post	Saving images to GoogleDrive using GoogleAPI, GoogleAppScript[GAS]	-	-	Watch videos on smartphone **Posting images to LINE with ESP32** this time

1-2. Camera use with ESP32 (price details)

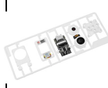
*Please note that the cost varies depending on the season.
* Excel can be downloaded from the Hobby-IT site.

①ESP32(WROOM) and OV2640 【¥3930】

NO	Item	quanti	Image	Item	URL(Japanese Shop)	Price(yen)	Note
1	ESP32 development board	1		ESP32-DevKitC-32E ESP32-WROOM-32E development board 4MB	https://akizukidenshi.com/catalog/g/gM-15673/	1600	19Pin x 2 rows
2	Breadboard 6 hole [EIC-3901]	1		Breadboard 6 hole plate EIC-3901	https://akizukidenshi.com/catalog/g/gP-12366/	460	
3	Green LED	1		3mm yellow-green LED 570nm 70 degrees OSG8HA3Z74A	https://akizukidenshi.com/catalog/g/gI-11637/	10	For status display
4	OV2640 camera module	1		2 megapixel camera using OV2640 B0011	https://akizukidenshi.com/catalog/g/gM-13197/	1680	
5	jumper cable	1		Cable with connector 20cm 40P male/female	https://akizukidenshi.com/catalog/g/gC-17228/	180	This time I used a handheld, so I haven't checked the connector shape etc.
total						3,930	**Separate shipping fee is required**

I omitted the jumper wire set and the LED resistor for wiring.

②M5Stack UnitCam 【¥4046】

NO	Item	quanti	Image	Item	Shop-URL		Price	Note	
1	UnitCam	1		Unit Cam Wi-Fi Camera DIY Kit (OV2640)	https://shop.m5stack.com/collections/m5-cameras	M5Stack	$18.95		
					https://www.marutsu.co.jp/pc/i/2228284/	marutsu	2946		
2	USB serial conversion module	1		FT234X ultra-compact USB serial conversion module	https://akizukidenshi.com/catalog/g/gM-08461/	AKIZUKI	680	Even if you use many UnitCams, you only need one	Required for writing software
3	Thin pin header 1 x 20	1		Thin pin header 1 x 20	https://akizukidenshi.com/catalog/g/gC-04398/		20		
4	jumper cable	1		Cable with connector 20cm 40P male/female	https://akizukidenshi.com/catalog/g/gC-17228/		180	This time I used a handheld, so I haven't checked	
5	bread board	1		bread board BB-801	https://akizukidenshi.com/catalog/g/gP-05294/		220		
total							4,046	**Separate shipping fee is required**	

There is also a dedicated Uploader, but this item is selected this time because it is versatile

③ESP32(WROVER) and OV2640 【¥4080】

NO	Item	quanti	Image	Item	URL(Japanese Shop)	Price(yen)	Note
1	ESP32 development board	1		ESP32-DevKitC-32E ESP32-WROVER-32E development board 8MB	https://akizukidenshi.com/catalog/g/gM-15674/	1750	19Pin x 2 rows
2	Breadboard 6 hole [EIC-3901]	1		Breadboard 6 hole plate EIC-3901	https://akizukidenshi.com/catalog/g/gP-12366/	460	
3	Green LED	1		3mm yellow-green LED 570nm 70 degrees OSG8HA3Z74A	https://akizukidenshi.com/catalog/g/gI-11637/	10	For status display
4	OV2640 camera module	1		2 megapixel camera using OV2640 B0011	https://akizukidenshi.com/catalog/g/gM-13197/	1680	
5	jumper cable	1		Cable with connector 20cm 40P male/female	https://akizukidenshi.com/catalog/g/gC-17228/	180	This time I used a handheld, so I haven't checked the connector shape etc.
total						4,080	**Separate shipping fee is required**

I omitted the jumper wire set and the LED resistor for wiring.

④M5Stack TimerCamera(OV3660) 【¥2596/2860】

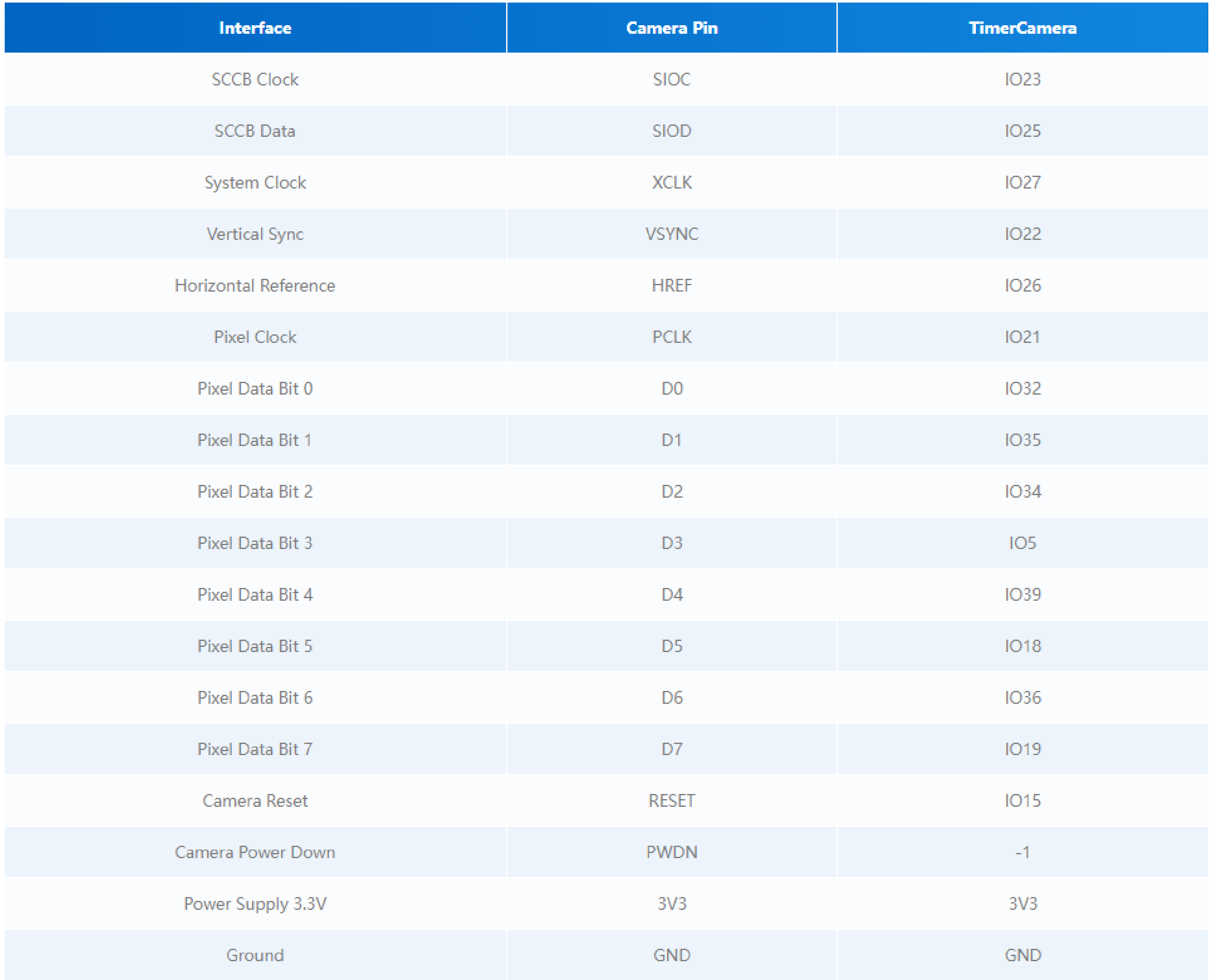
NO	Item	quanti	Image	Item	Shop-URL		Price	Note
1	Timer Camera X	1		ESP32 PSRAM Timer Camera X (OV3660)	https://shop.m5stack.com/collections/m5-cameras	M5Stack	$17.95	viewing angle 66.5°
					https://www.switch-science.com/products/6742	SWITCH SIENCE	2596	
1	Timer Camera F	1		ESP32 PSRAM Timer Camera F (OV3660)	https://shop.m5stack.com/collections/m5-cameras	M5Stack	$18.95	viewing angle 120°
					https://www.switch-science.com/products/6786	SWITCH SIENCE	2860	
Total							2,860	**Separate shipping fee is required**

X/F is the difference in viewing angle.
Development is possible with a personal computer with a micro USB cable.

1-3. TimerCamera

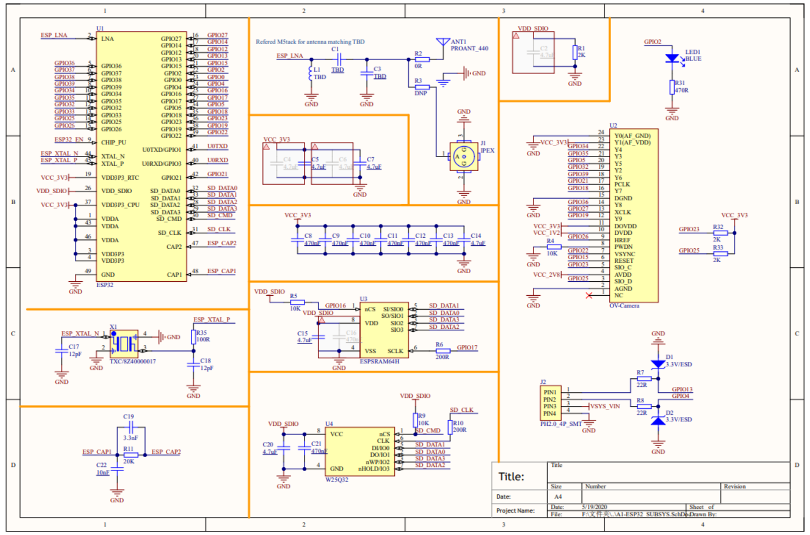
●M5Stack Official TimerCamera Document
https://docs.m5stack.com/en/unit/timercam_x

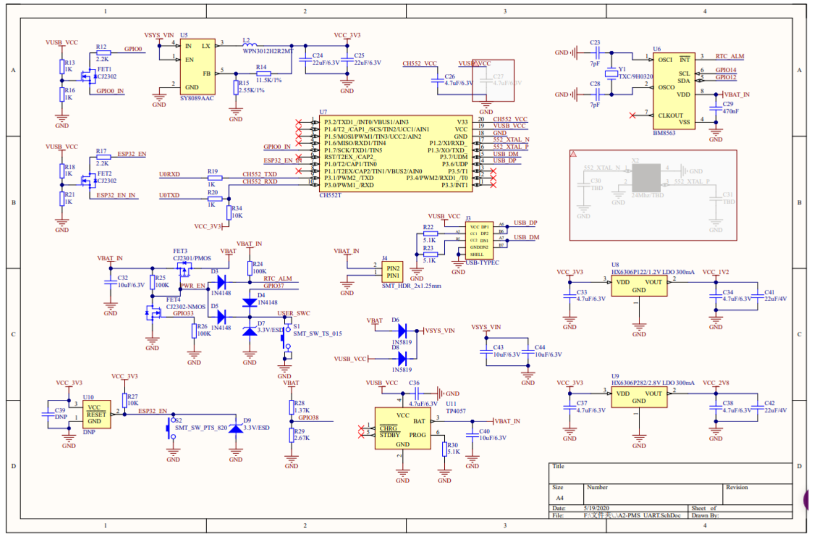
- Pin Map



Interface	Camera Pin	TimerCamera
SCCB Clock	SIOC	IO23
SCCB Data	SIOD	IO25
System Clock	XCLK	IO27
Vertical Sync	VSYNC	IO22
Horizontal Reference	HREF	IO26
Pixel Clock	PCLK	IO21
Pixel Data Bit 0	D0	IO32
Pixel Data Bit 1	D1	IO35
Pixel Data Bit 2	D2	IO34
Pixel Data Bit 3	D3	IO5
Pixel Data Bit 4	D4	IO39
Pixel Data Bit 5	D5	IO18
Pixel Data Bit 6	D6	IO36
Pixel Data Bit 7	D7	IO19
Camera Reset	RESET	IO15
Camera Power Down	PWDN	-1
Power Supply 3.3V	3V3	3V3
Ground	GND	GND

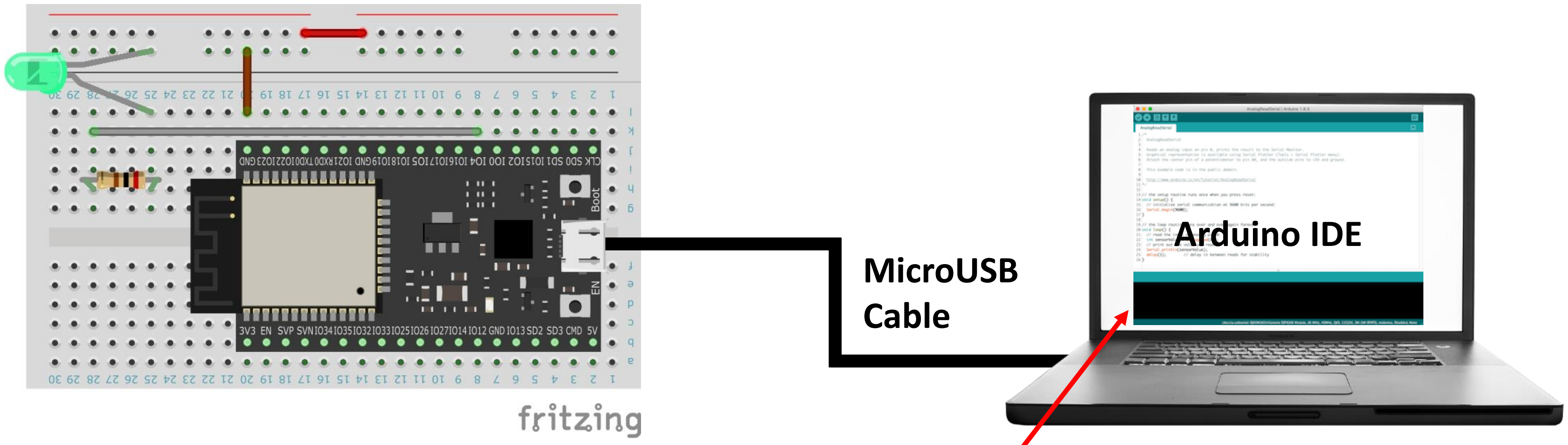
- Schematic





2. the development environment “Arduino”

We will use Arduino as the development environment.



ESP32 Development Board

MicroUSB Cable

Arduino IDE

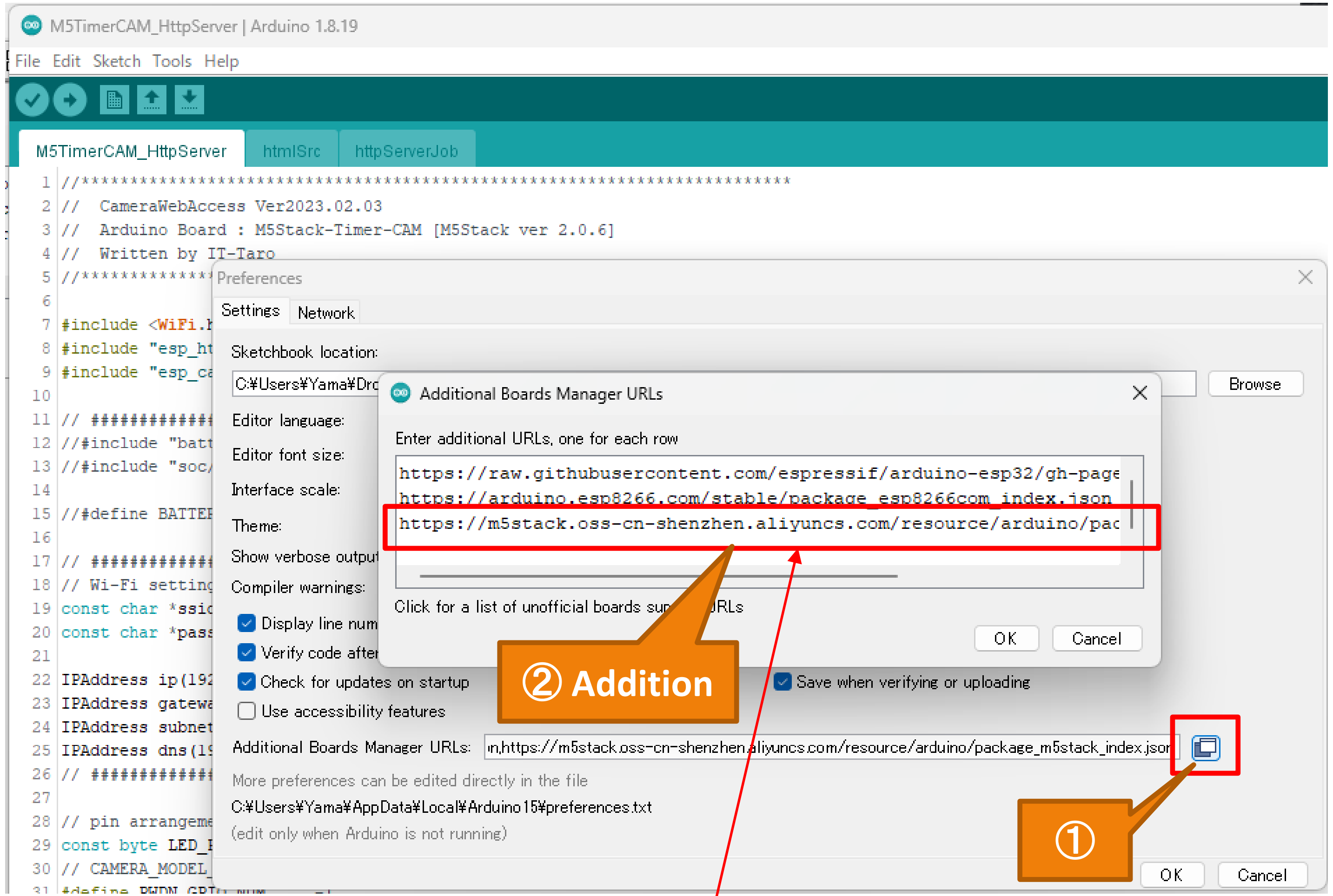
P C (Windows/Mac/Linux)

【Arduino Official site】
https://www.arduino.cc/
Downloadable

3-1. Arduino settings (Board settings)

●M5Stack Official　ArduinoIDE Setting
https://docs.m5stack.com/en/quick_start/timer_cam/arduino

1) Add Additional Board Manager setting from ArduinoIDE setting


M5TimerCAM_HttpServer | Arduino 1.8.19
File Edit Sketch Tools Help
M5TimerCAM_HttpServer
htmlSrc
httpServerJob

```
//*********************************************************************
//  CameraWebAccess Ver2023.02.03
//  Arduino Board : M5Stack-Timer-CAM [M5Stack ver 2.0.6]
//  Written by IT-Taro
//*************

#include <WiFi.h
#include "esp_ht
#include "esp_ca

// ############
//#include "batt
//#include "soc/

//#define BATTER

// ############
// Wi-Fi setting
const char *ssid
const char *pass

IPAddress ip(192
IPAddress gatewa
IPAddress subnet
IPAddress dns(19
// ############

// pin arrangeme
const byte LED_P
// CAMERA_MODEL_
#define PWDN_GPIO_NUM
```

Preferences

Settings　Network

Sketchbook location: C:¥Users¥Yama¥Dro　Browse

Editor language:

Editor font size:

Interface scale:

Theme:

Show verbose output

Compiler warnings:

☑ Display line num

☑ Verify code after

☑ Check for updates on startup

☐ Use accessibility features

☑ Save when verifying or uploading

Additional Boards Manager URLs: n,https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/arduino/package_m5stack_index.json

More preferences can be edited directly in the file

C:¥Users¥Yama¥AppData¥Local¥Arduino15¥preferences.txt

(edit only when Arduino is not running)

OK　Cancel

Additional Boards Manager URLs

Enter additional URLs, one for each row

```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-page
https://arduino.esp8266.com/stable/package_esp8266com_index.json
https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/arduino/pac
```

Click for a list of unofficial boards sup　URLs

OK　Cancel

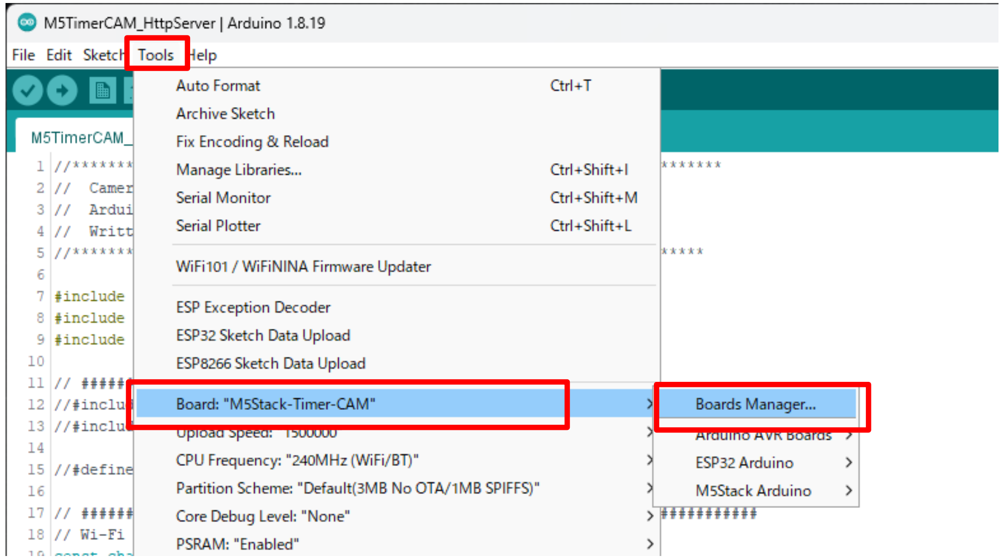
② Addition

①

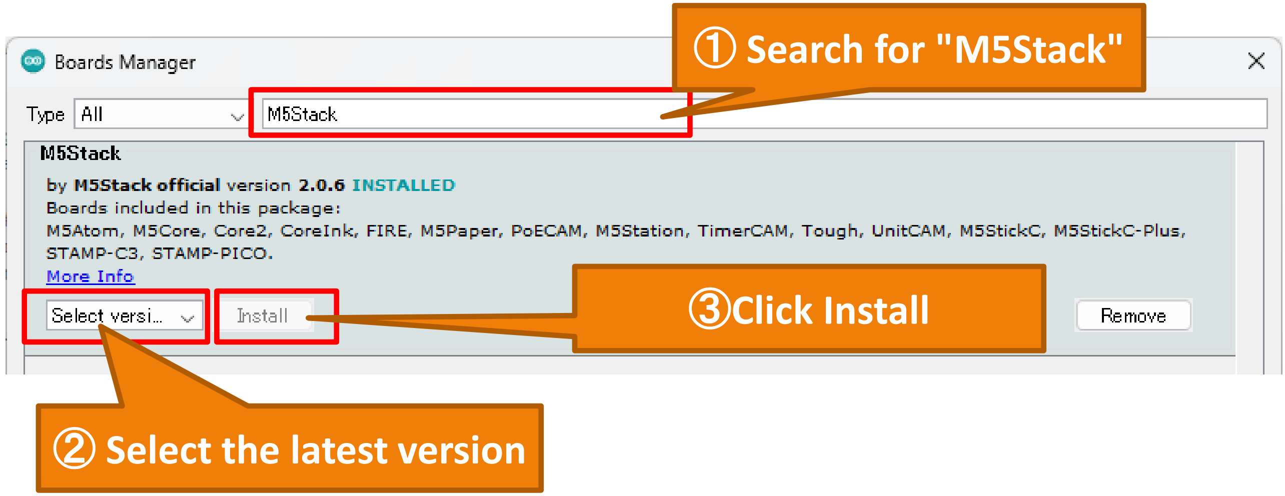
set value：

https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/arduino/package_m5stack_index.json

2) Launch Board Manager


M5TimerCAM_HttpServer | Arduino 1.8.19
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
ESP Exception Decoder
ESP32 Sketch Data Upload
ESP8266 Sketch Data Upload
Board: "M5Stack-Timer-CAM"
Upload Speed: "1500000"
CPU Frequency: "240MHz (WiFi/BT)"
Partition Scheme: "Default(3MB No OTA/1MB SPIFFS)"
Core Debug Level: "None"
PSRAM: "Enabled"
Boards Manager...
Arduino AVR Boards
ESP32 Arduino
M5Stack Arduino

3) Install M5Stack



M5Stack

by **M5Stack official** version **2.0.6** **INSTALLED**

Boards included in this package:

M5Atom, M5Core, Core2, CoreInk, FIRE, M5Paper, PoECAM, M5Station, TimerCAM, Tough, UnitCAM, M5StickC, M5StickC-Plus, STAMP-C3, STAMP-PICO.

More Info

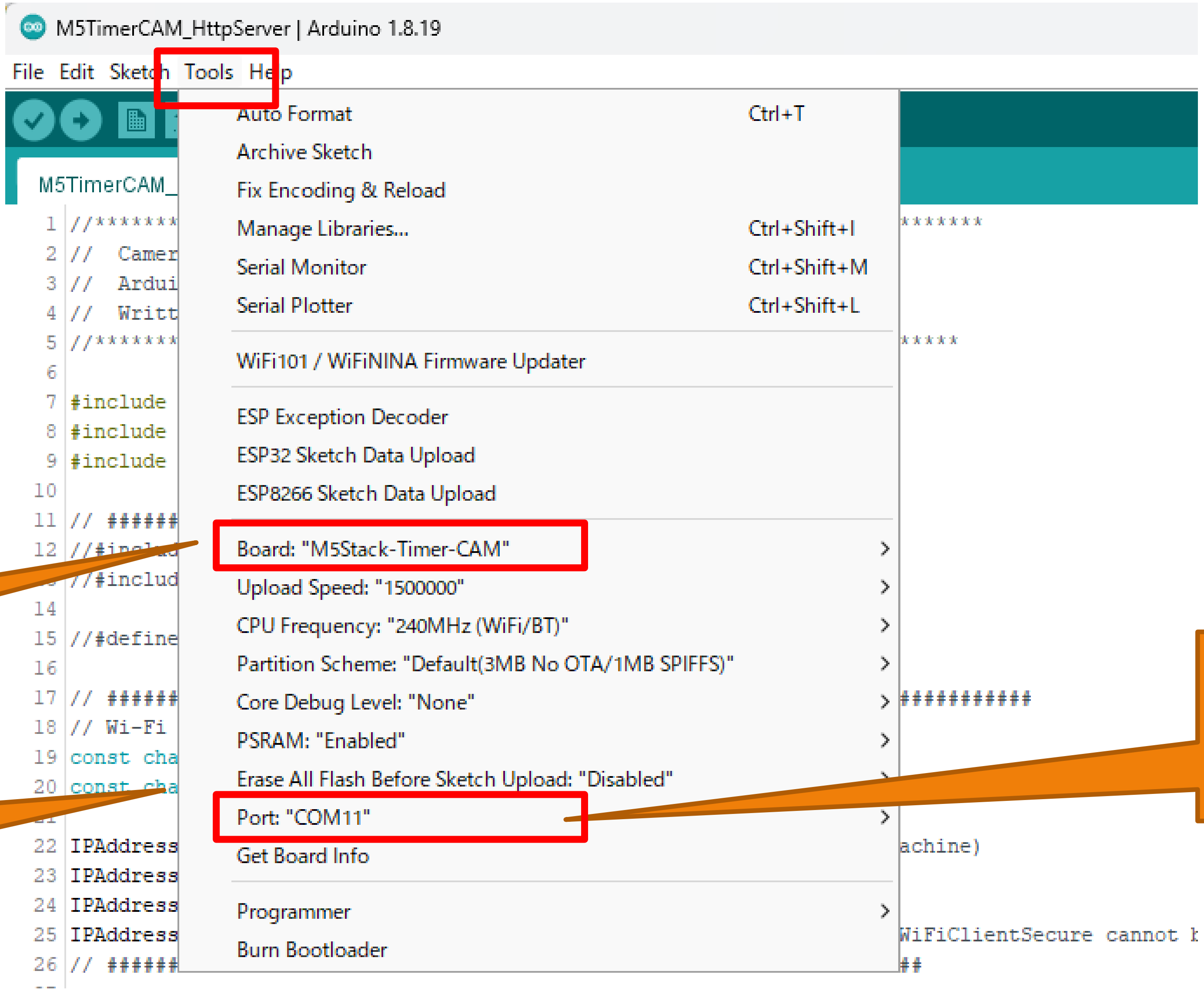
① Search for "M5Stack"

②Select the latest version

③Click Install

3-1. Arduino settings (Board settings)

4) Set Board to "M5Stack-Timer-CAM"



M5TimerCAM_HttpServer | Arduino 1.8.19

```
//*******                                                  *******
//  Camer
//  Ardui
//  Writt
//*******                                                  *****

#include
#include
#include

// ######
//#includ
//#includ

//#define

// ######                                                  ###########
// Wi-Fi
const cha
const cha

IPAddress                                                  achine)
IPAddress
IPAddress
IPAddress                                                  WiFiClientSecure cannot k
// ######                                                  ##
```

Auto Format	Ctrl+T
Archive Sketch	
Fix Encoding & Reload	
Manage Libraries...	Ctrl+Shift+I
Serial Monitor	Ctrl+Shift+M
Serial Plotter	Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater	
ESP Exception Decoder	
ESP32 Sketch Data Upload	
ESP8266 Sketch Data Upload	
Board: "M5Stack-Timer-CAM"	›
Upload Speed: "1500000"	›
CPU Frequency: "240MHz (WiFi/BT)"	›
Partition Scheme: "Default(3MB No OTA/1MB SPIFFS)"	›
Core Debug Level: "None"	›
PSRAM: "Enabled"	›
Erase All Flash Before Sketch Upload: "Disabled"	›
Port: "COM11"	›
Get Board Info	
Programmer	›
Burn Bootloader	

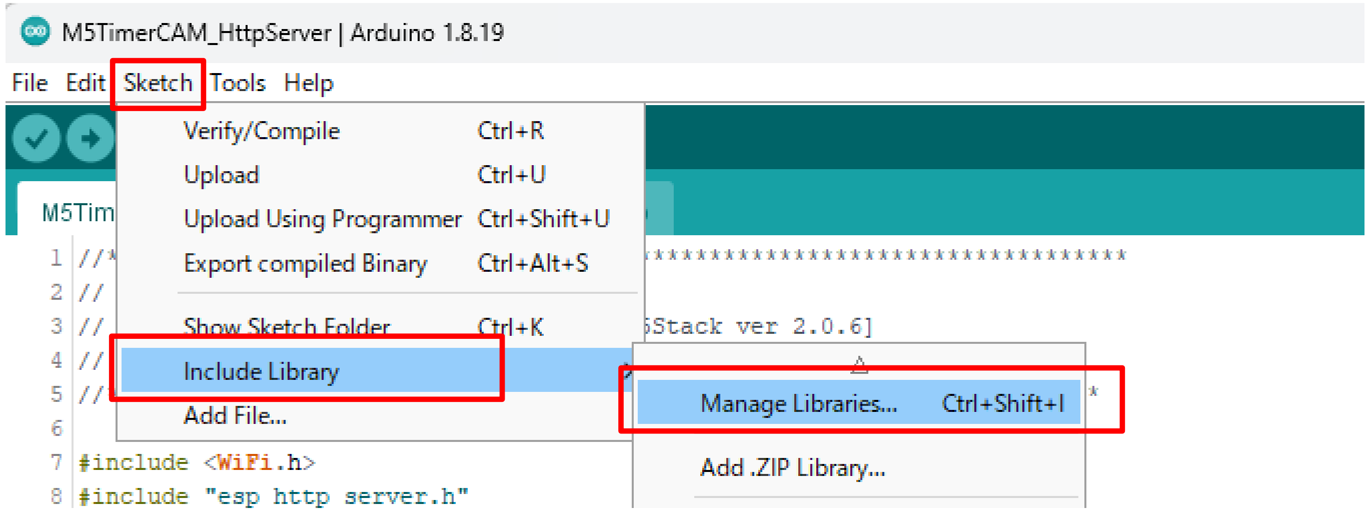
Select "M5Stack-Timer-CAM"

No other settings changed (with default value)

For Port, select the port to which TimerCamera is connected [Write error when selection fails]

3-2. Arduino settings (Add Library)

1) Start Library Manager



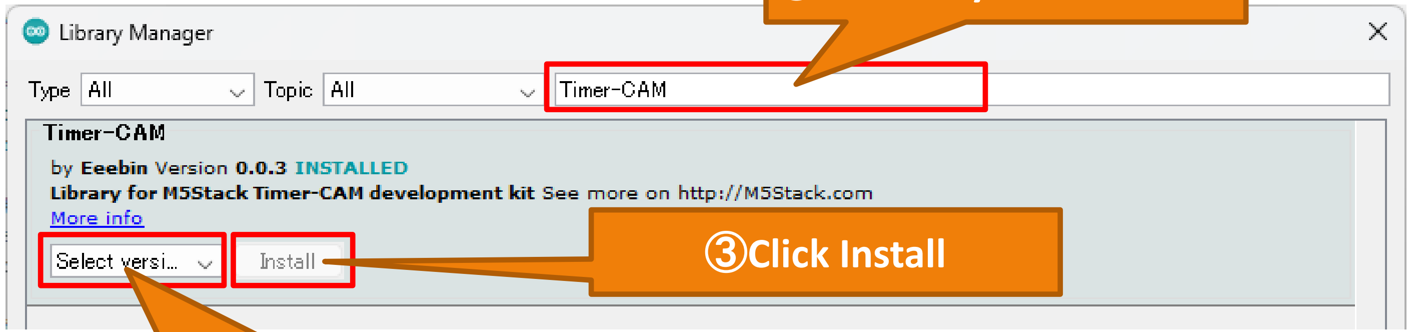
M5TimerCAM_HttpServer | Arduino 1.8.19

File
Edit
Sketch
Tools
Help
M5Tim
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...

```
//*                                    ***********************************
//
//                                    5Stack ver 2.0.6]
//
//

#include <WiFi.h>
#include "esp http server.h"
```

2) Install "Timer-CAM"

①Search by "Timer-CAM"



Library Manager

Type All Topic All Timer-CAM

Timer-CAM

by **Eeebin** Version **0.0.3** **INSTALLED**

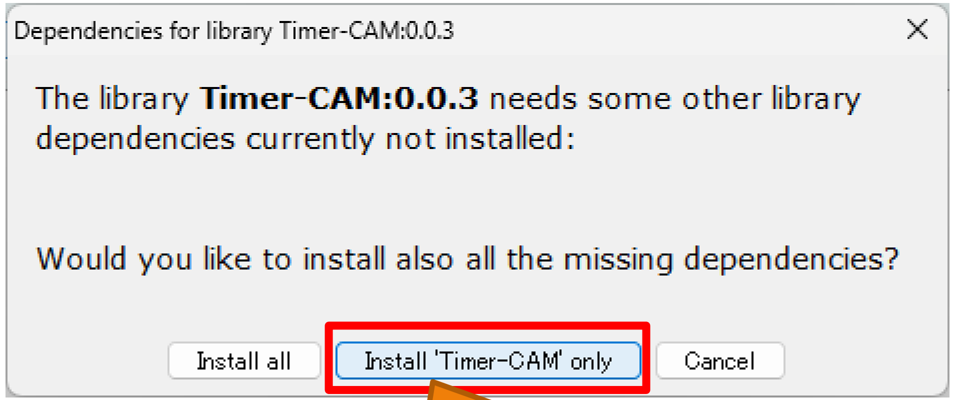
Library for M5Stack Timer-CAM development kit See more on http://M5Stack.com

More info

③Click Install

② Select the latest version

3) Install only "Timer-CAM"



Dependencies for library Timer-CAM:0.0.3

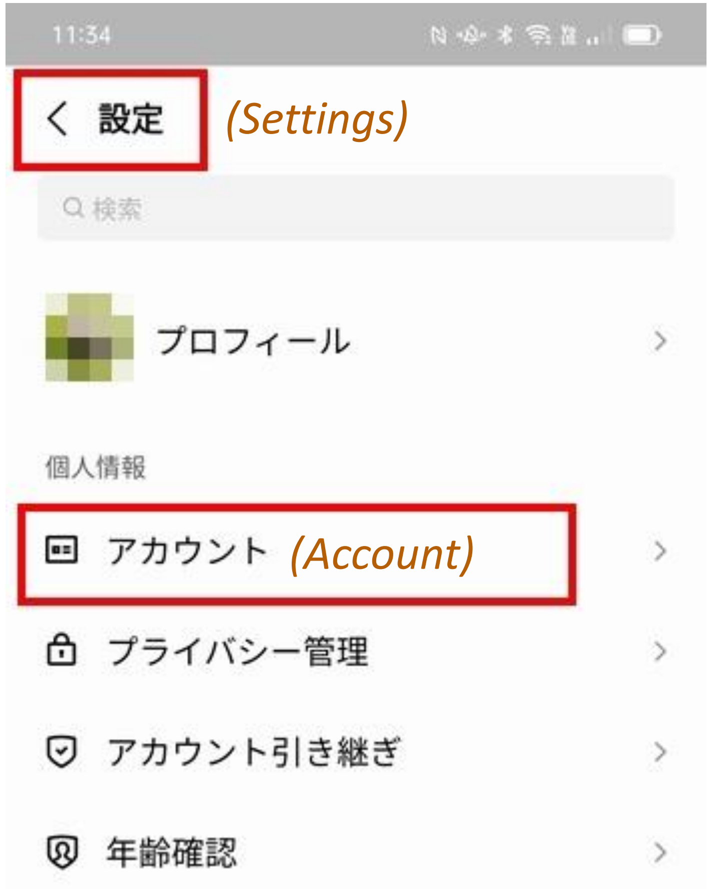
The library **Timer-CAM:0.0.3** needs some other library dependencies currently not installed:

Would you like to install also all the missing dependencies?

just install this
(When operation is NG, install all)

4-1. LINE settings (token acquisition)

1) Select an account from "Settings" on the smartphone LINE app



11:34

< 設定 *(Settings)*

検索

プロフィール

個人情報

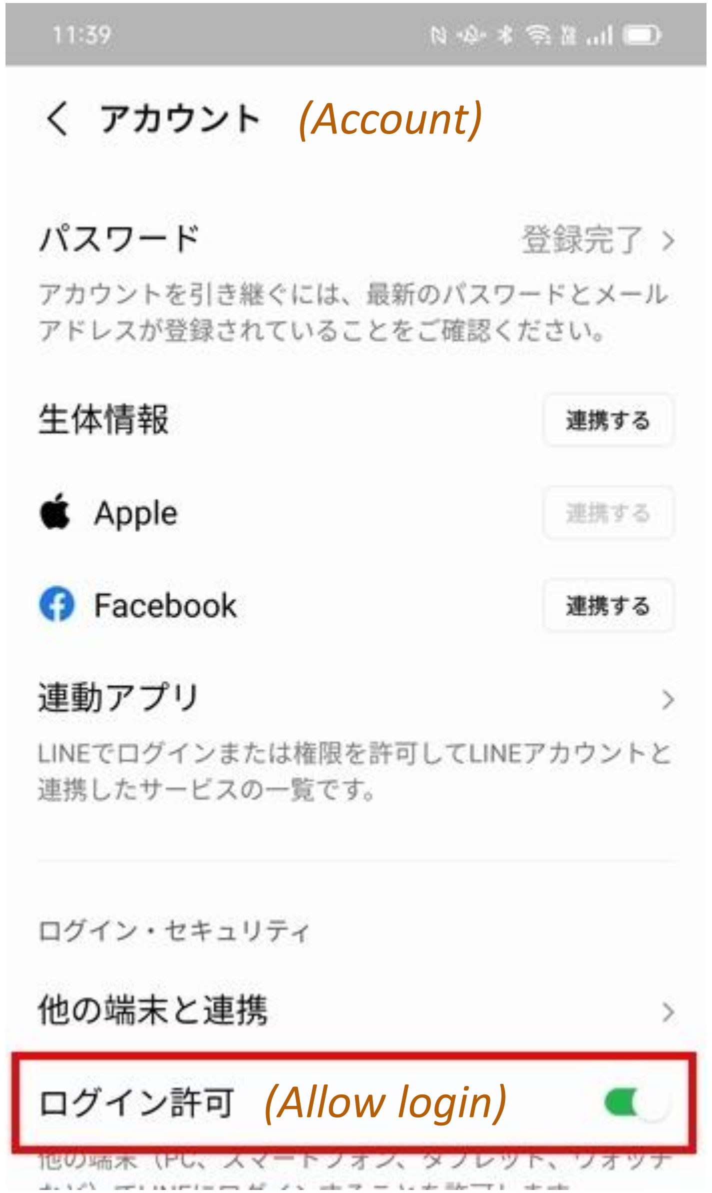
アカウント *(Account)*

プライバシー管理

アカウント引き継ぎ

年齢確認

2) Turn on “Allow login"



11:39

< アカウント *(Account)*

パスワード 登録完了 >

アカウントを引き継ぐには、最新のパスワードとメールアドレスが登録されていることをご確認ください。

生体情報 連携する

Apple 連携する

Facebook 連携する

連動アプリ

LINEでログインまたは権限を許可してLINEアカウントと連携したサービスの一覧です。

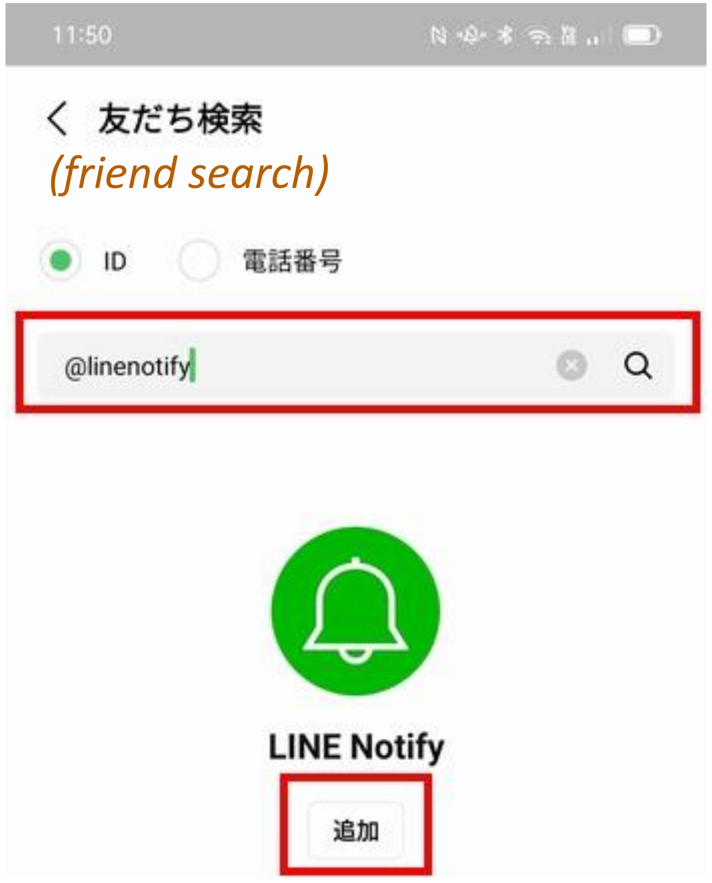
ログイン・セキュリティ

他の端末と連携

ログイン許可 *(Allow login)*

他の端末（PC、スマートフォン、タブレット、ウォッチなど）でLINEにログインすることを許可します。

3) Add friends by searching for "@linenotify"



11:50

< 友だち検索

(friend search)

● ID ○ 電話番号

@linenotify

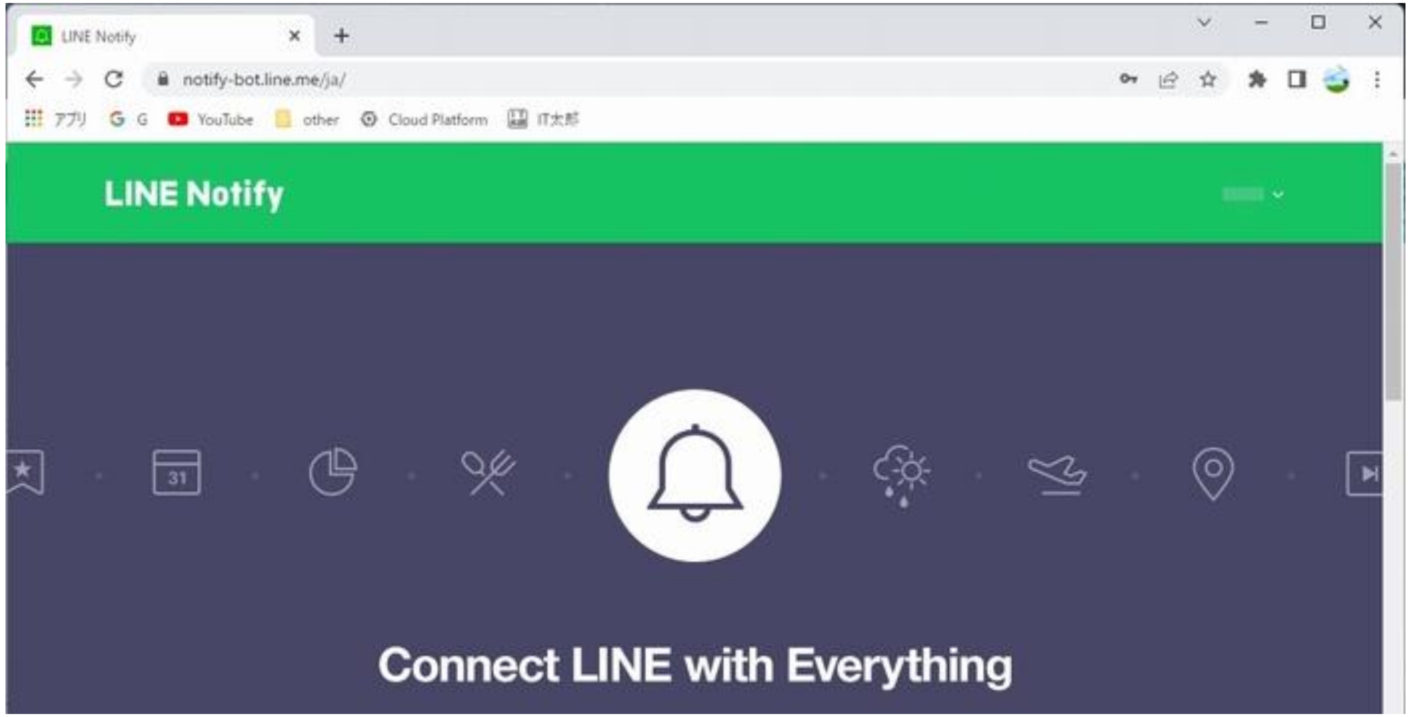
LINE Notify

追加

(Add)

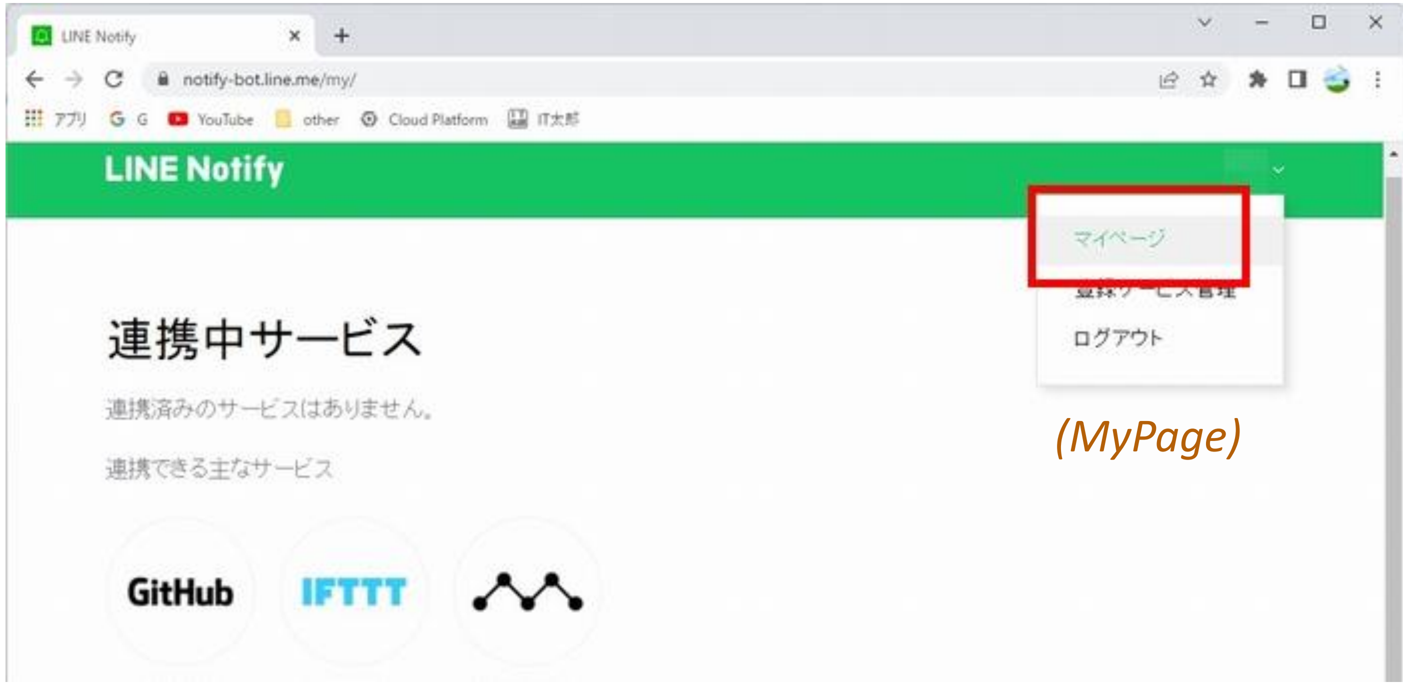
4-2. LINE settings (token acquisition)

4) Access LINE Notify on your computer and log in with your LINE account.



https://notify-bot.line.me/ja/

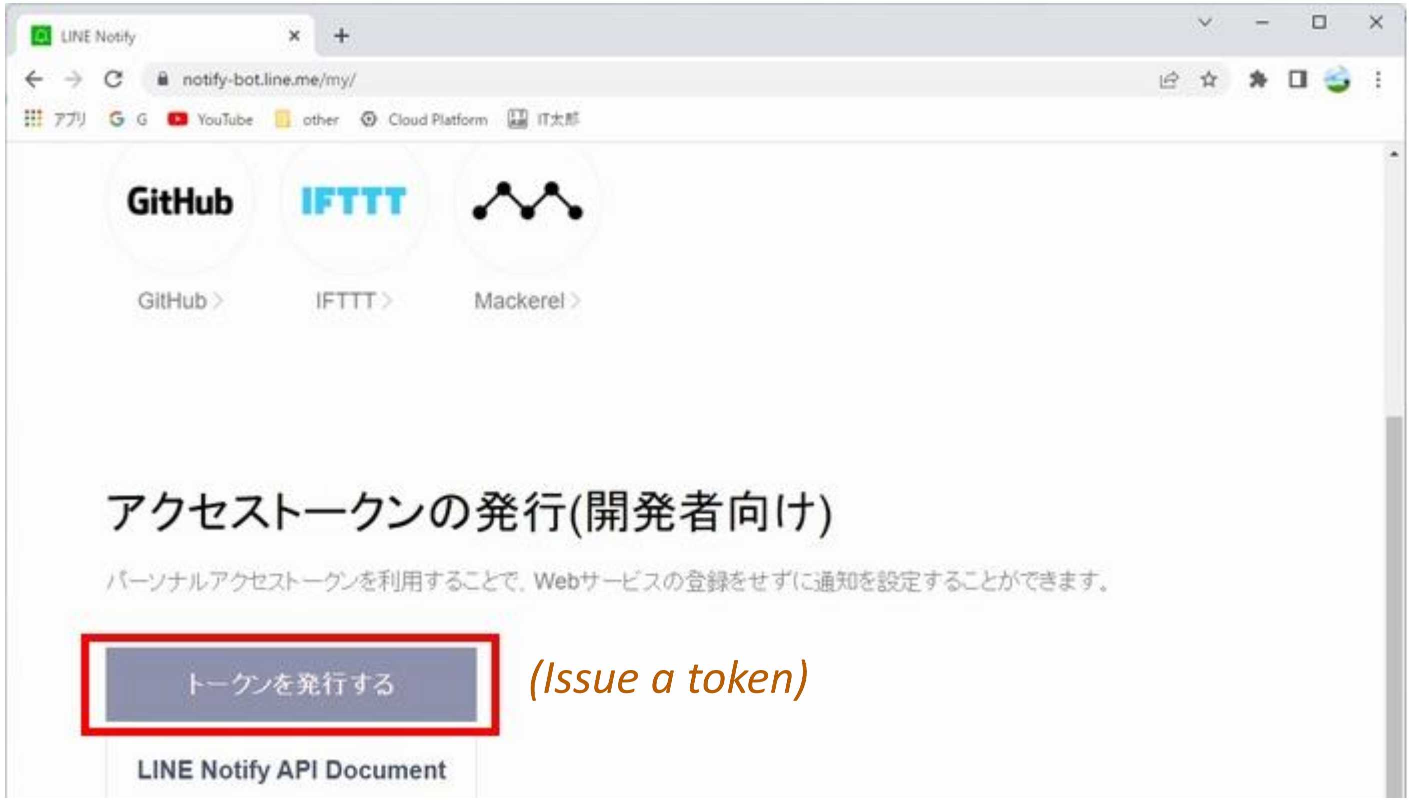
5) Select "My Page" from the menu on the upper right



(MyPage)

4-3. LINE settings (token acquisition)

6) Issue a token.

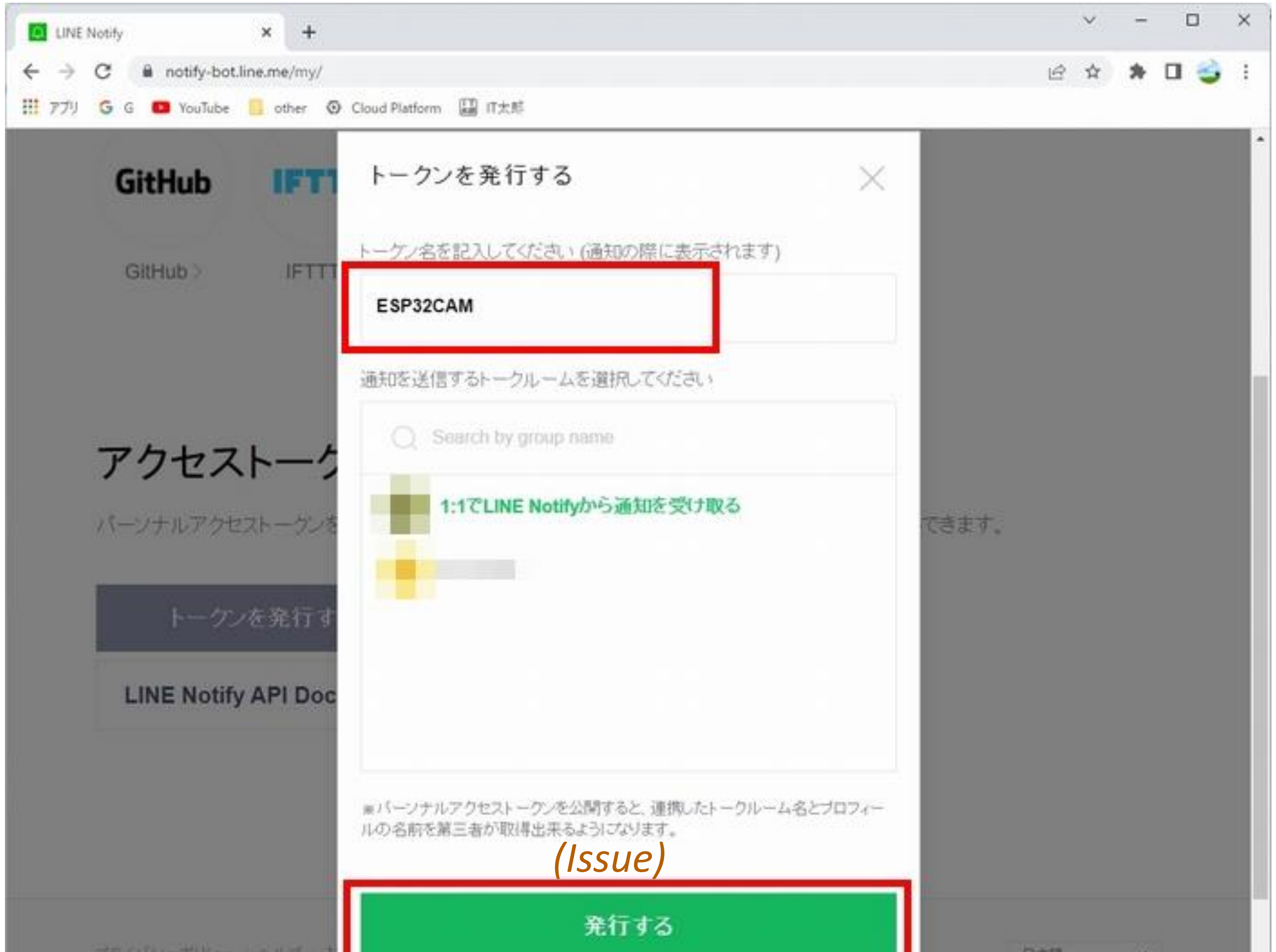


アクセストークンの発行(開発者向け)

パーソナルアクセストークンを利用することで、Webサービスの登録をせずに通知を設定することができます。

(Issue a token)

7) Enter the token name and issue.



トークンを発行する

トークン名を記入してください (通知の際に表示されます)

ESP32CAM

通知を送信するトークルームを選択してください

1:1でLINE Notifyから通知を受け取る

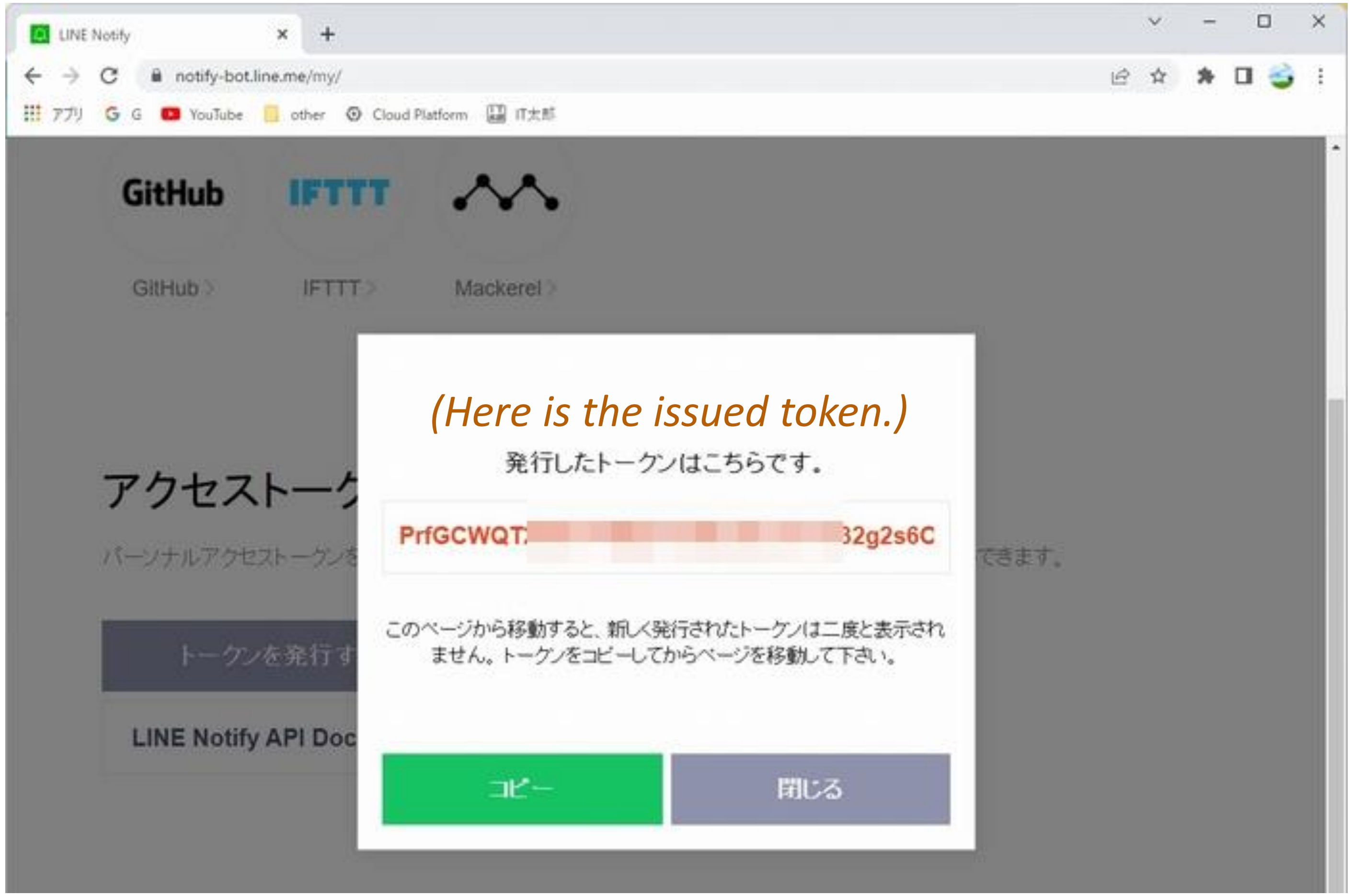
[redacted]

※パーソナルアクセストークンを公開すると、連携したトークルーム名とプロフィールの名前を第三者が取得出来るようになります。

(Issue)

4-4. LINE settings (token acquisition)

8) Get the issued token. (Recorded for inclusion in the program.)



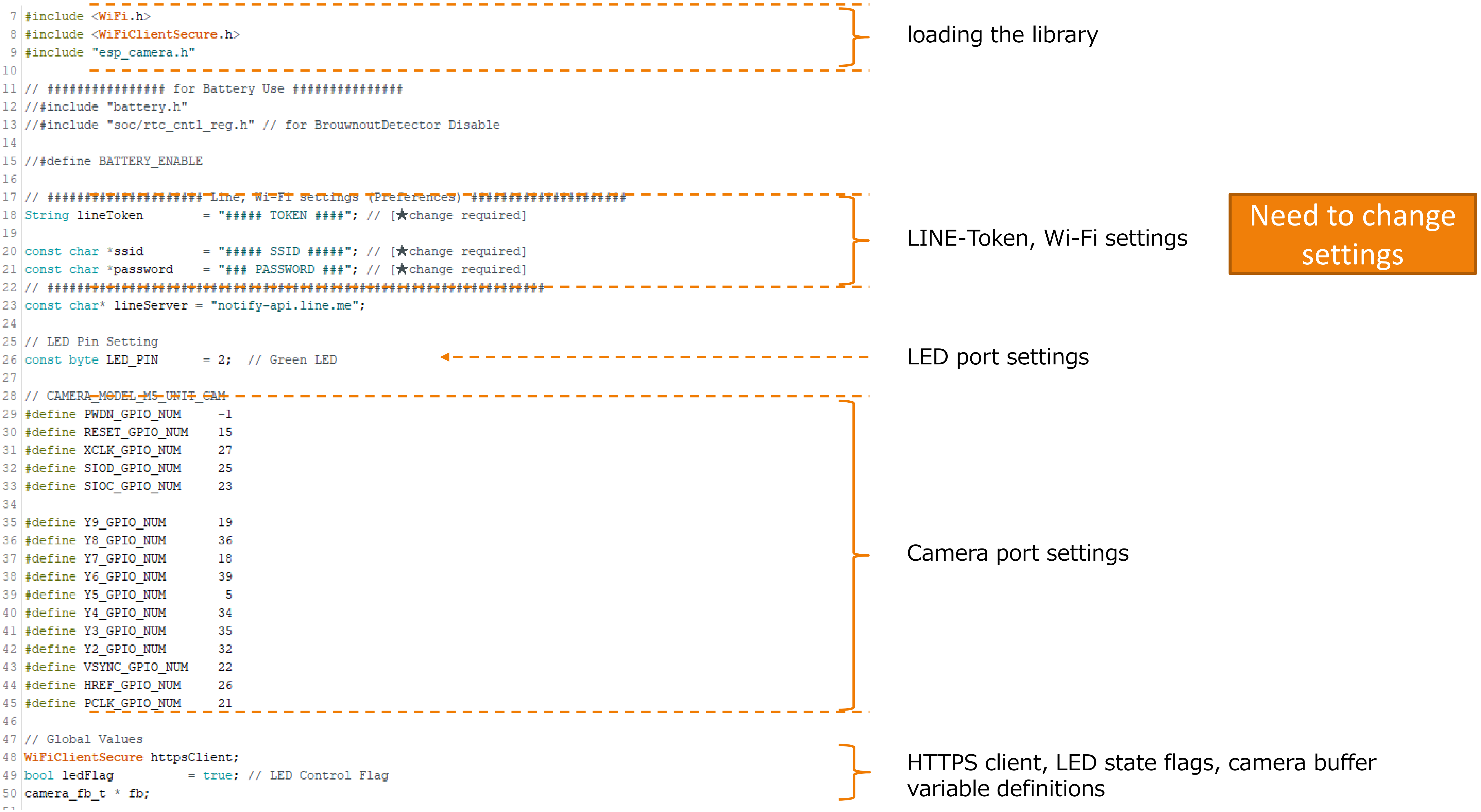
(Here is the issued token.)

発行したトークンはこちらです。

PrfGCWQT[redacted]32g2s6C

このページから移動すると、新しく発行されたトークンは二度と表示されません。トークンをコピーしてからページを移動して下さい。

5. Arduino program (global definition)



```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "esp_camera.h"

// ################ for Battery Use ###############
//#include "battery.h"
//#include "soc/rtc_cntl_reg.h" // for BrounoutDetector Disable

//#define BATTERY_ENABLE

// ##################### Line, Wi-Fi settings (Preferences) #####################
String lineToken        = "##### TOKEN ####"; // [★change required]

const char *ssid        = "##### SSID #####"; // [★change required]
const char *password    = "### PASSWORD ###"; // [★change required]
// ####################################################################
const char* lineServer = "notify-api.line.me";

// LED Pin Setting
const byte LED_PIN      = 2;  // Green LED

// CAMERA_MODEL_M5_UNIT_CAM
#define PWDN_GPIO_NUM      -1
#define RESET_GPIO_NUM    15
#define XCLK_GPIO_NUM     27
#define SIOD_GPIO_NUM     25
#define SIOC_GPIO_NUM     23

#define Y9_GPIO_NUM       19
#define Y8_GPIO_NUM       36
#define Y7_GPIO_NUM       18
#define Y6_GPIO_NUM       39
#define Y5_GPIO_NUM        5
#define Y4_GPIO_NUM       34
#define Y3_GPIO_NUM       35
#define Y2_GPIO_NUM       32
#define VSYNC_GPIO_NUM    22
#define HREF_GPIO_NUM     26
#define PCLK_GPIO_NUM     21

// Global Values
WiFiClientSecure httpsClient;
bool ledFlag          = true; // LED Control Flag
camera_fb_t * fb;
```

loading the library

LINE-Token, Wi-Fi settings

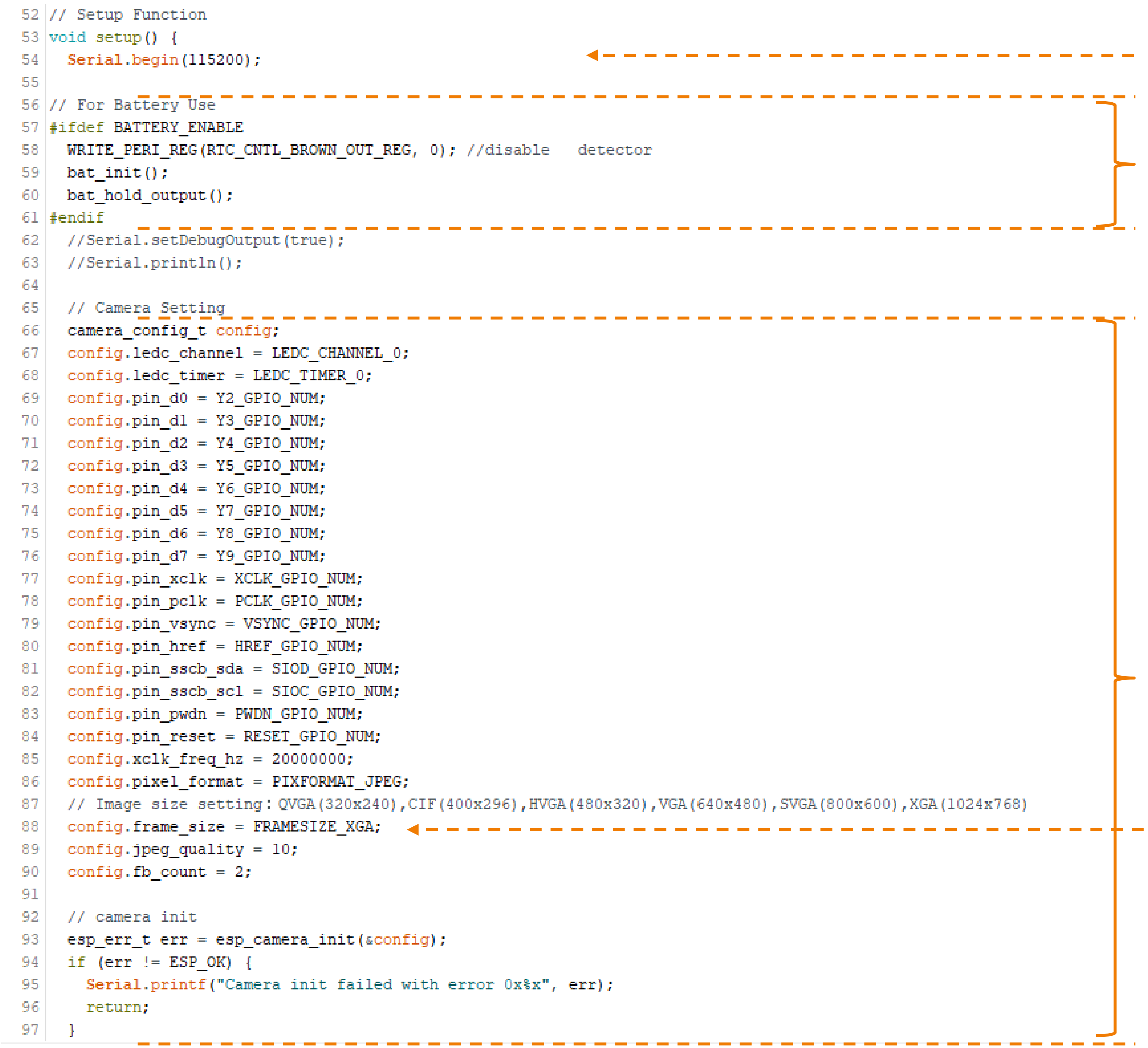
Need to change settings

LED port settings

Camera port settings

HTTPS client, LED state flags, camera buffer variable definitions

5. Arduino program (Setup function)



```
// Setup Function
void setup() {
  Serial.begin(115200);

// For Battery Use
#ifdef BATTERY_ENABLE
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //disable   detector
  bat_init();
  bat_hold_output();
#endif
  //Serial.setDebugOutput(true);
  //Serial.println();

  // Camera Setting
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
  // Image size setting: QVGA(320x240),CIF(400x296),HVGA(480x320),VGA(640x480),SVGA(800x600),XGA(1024x768)
  config.frame_size = FRAMESIZE_XGA;
  config.jpeg_quality = 10;
  config.fb_count = 2;

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }
```

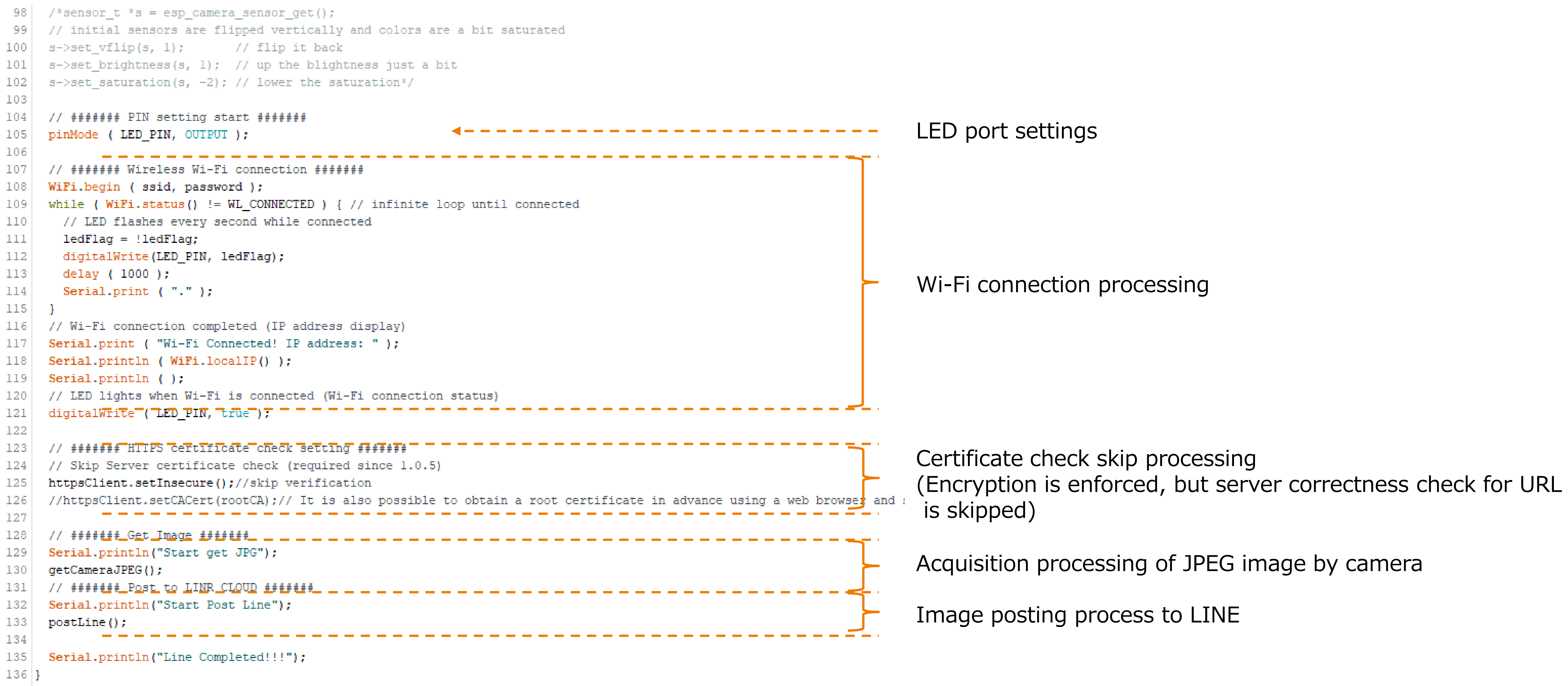
Start serial monitor

Turn off BrownOUT check when using battery
[Enable setting if an error occurs immediately and restarts]

Camera port settings and initialization

Image size setting
XGA(1024*768)

5. Arduino program (Setup function)



```
  /*sensor_t *s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit saturated
  s->set_vflip(s, 1);       // flip it back
  s->set_brightness(s, 1);  // up the blightness just a bit
  s->set_saturation(s, -2); // lower the saturation*/

  // ####### PIN setting start #######
  pinMode ( LED_PIN, OUTPUT );

  // ####### Wireless Wi-Fi connection #######
  WiFi.begin ( ssid, password );
  while ( WiFi.status() != WL_CONNECTED ) { // infinite loop until connected
    // LED flashes every second while connected
    ledFlag = !ledFlag;
    digitalWrite(LED_PIN, ledFlag);
    delay ( 1000 );
    Serial.print ( "." );
  }
  // Wi-Fi connection completed (IP address display)
  Serial.print ( "Wi-Fi Connected! IP address: " );
  Serial.println ( WiFi.localIP() );
  Serial.println ( );
  // LED lights when Wi-Fi is connected (Wi-Fi connection status)
  digitalWrite ( LED_PIN, true );

  // ####### HTTPS certificate check setting #######
  // Skip Server certificate check (required since 1.0.5)
  httpsClient.setInsecure();//skip verification
  //httpsClient.setCACert(rootCA);// It is also possible to obtain a root certificate in advance using a web browser and

  // ####### Get Image #######
  Serial.println("Start get JPG");
  getCameraJPEG();
  // ####### Post to LINR CLOUD #######
  Serial.println("Start Post Line");
  postLine();

  Serial.println("Line Completed!!!");
}
```

LED port settings

Wi-Fi connection processing

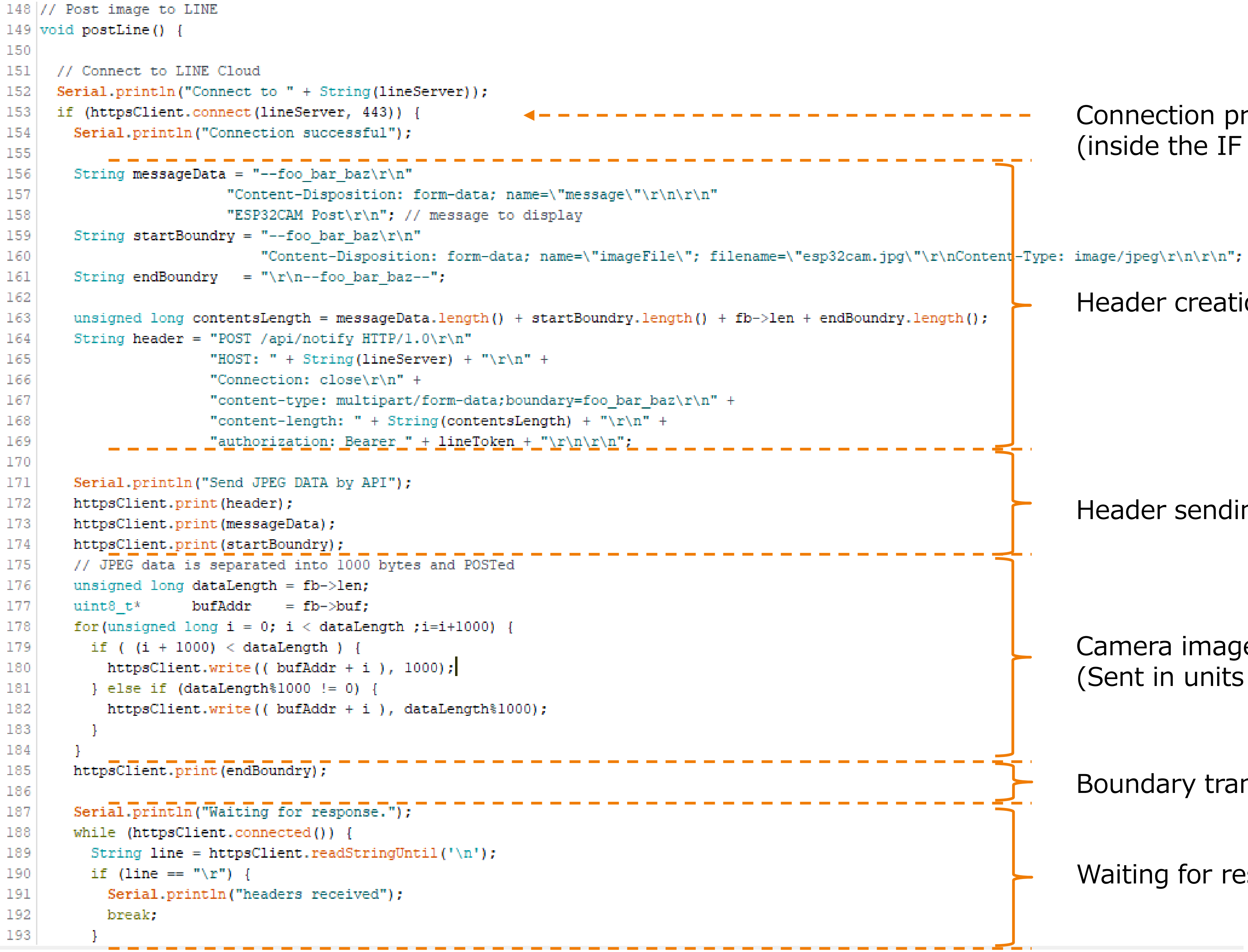
Certificate check skip processing
(Encryption is enforced, but server correctness check for URL is skipped)

Acquisition processing of JPEG image by camera

Image posting process to LINE

[reference]
Regarding the usage when the certificate is checked, it is implemented in the following post
"Smart Remote Controller (smartphone from outdoors, AI speaker cooperation) “
https://hobby-it.com/smartremo7/

5. Arduino program (postLine function)



```
// Post image to LINE
void postLine() {

  // Connect to LINE Cloud
  Serial.println("Connect to " + String(lineServer));
  if (httpsClient.connect(lineServer, 443)) {
    Serial.println("Connection successful");

    String messageData = "--foo_bar_baz\r\n"
                      "Content-Disposition: form-data; name=\"message\"\r\n\r\n"
                      "ESP32CAM Post\r\n"; // message to display
    String startBoundry = "--foo_bar_baz\r\n"
                          "Content-Disposition: form-data; name=\"imageFile\"; filename=\"esp32cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
    String endBoundry   = "\r\n--foo_bar_baz--";

    unsigned long contentsLength = messageData.length() + startBoundry.length() + fb->len + endBoundry.length();
    String header = "POST /api/notify HTTP/1.0\r\n"
                    "HOST: " + String(lineServer) + "\r\n" +
                    "Connection: close\r\n" +
                    "content-type: multipart/form-data;boundary=foo_bar_baz\r\n" +
                    "content-length: " + String(contentsLength) + "\r\n" +
                    "authorization: Bearer " + lineToken + "\r\n\r\n";

    Serial.println("Send JPEG DATA by API");
    httpsClient.print(header);
    httpsClient.print(messageData);
    httpsClient.print(startBoundry);
    // JPEG data is separated into 1000 bytes and POSTed
    unsigned long dataLength = fb->len;
    uint8_t*      bufAddr    = fb->buf;
    for(unsigned long i = 0; i < dataLength ;i=i+1000) {
      if ( (i + 1000) < dataLength ) {
        httpsClient.write(( bufAddr + i ), 1000);
      } else if (dataLength%1000 != 0) {
        httpsClient.write(( bufAddr + i ), dataLength%1000);
      }
    }
    httpsClient.print(endBoundry);

    Serial.println("Waiting for response.");
    while (httpsClient.connected()) {
      String line = httpsClient.readStringUntil('\n');
      if (line == "\r") {
        Serial.println("headers received");
        break;
      }
```

Connection processing to the server
(inside the IF statement if the connection is successful)

Header creation process

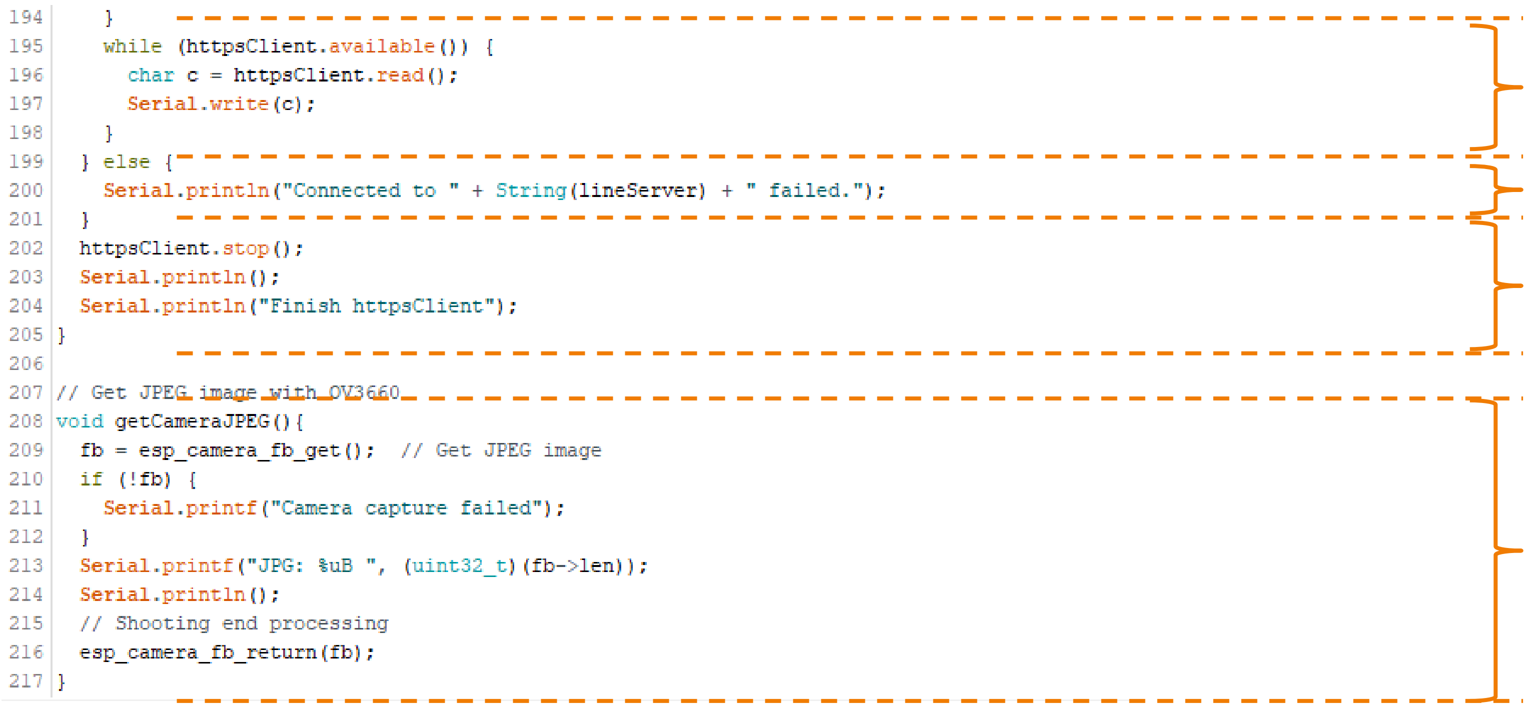
Header sending process

Camera image transmission processing
(Sent in units of 1000 bytes)

Boundary transmission of end of image transmission

Waiting for response from server (waiting for newline code)

5. Arduino program (postLine/getCameraJPEG function)



```
    }
    while (httpsClient.available()) {
      char c = httpsClient.read();
      Serial.write(c);
    }
  } else {
    Serial.println("Connected to " + String(lineServer) + " failed.");
  }
  httpsClient.stop();
  Serial.println();
  Serial.println("Finish httpsClient");
}

// Get JPEG image with OV3660
void getCameraJPEG(){
  fb = esp_camera_fb_get();  // Get JPEG image
  if (!fb) {
    Serial.printf("Camera capture failed");
  }
  Serial.printf("JPG: %uB ", (uint32_t)(fb->len));
  Serial.println();
  // Shooting end processing
  esp_camera_fb_return(fb);
}
```

Display the contents received from the server on the serial monitor

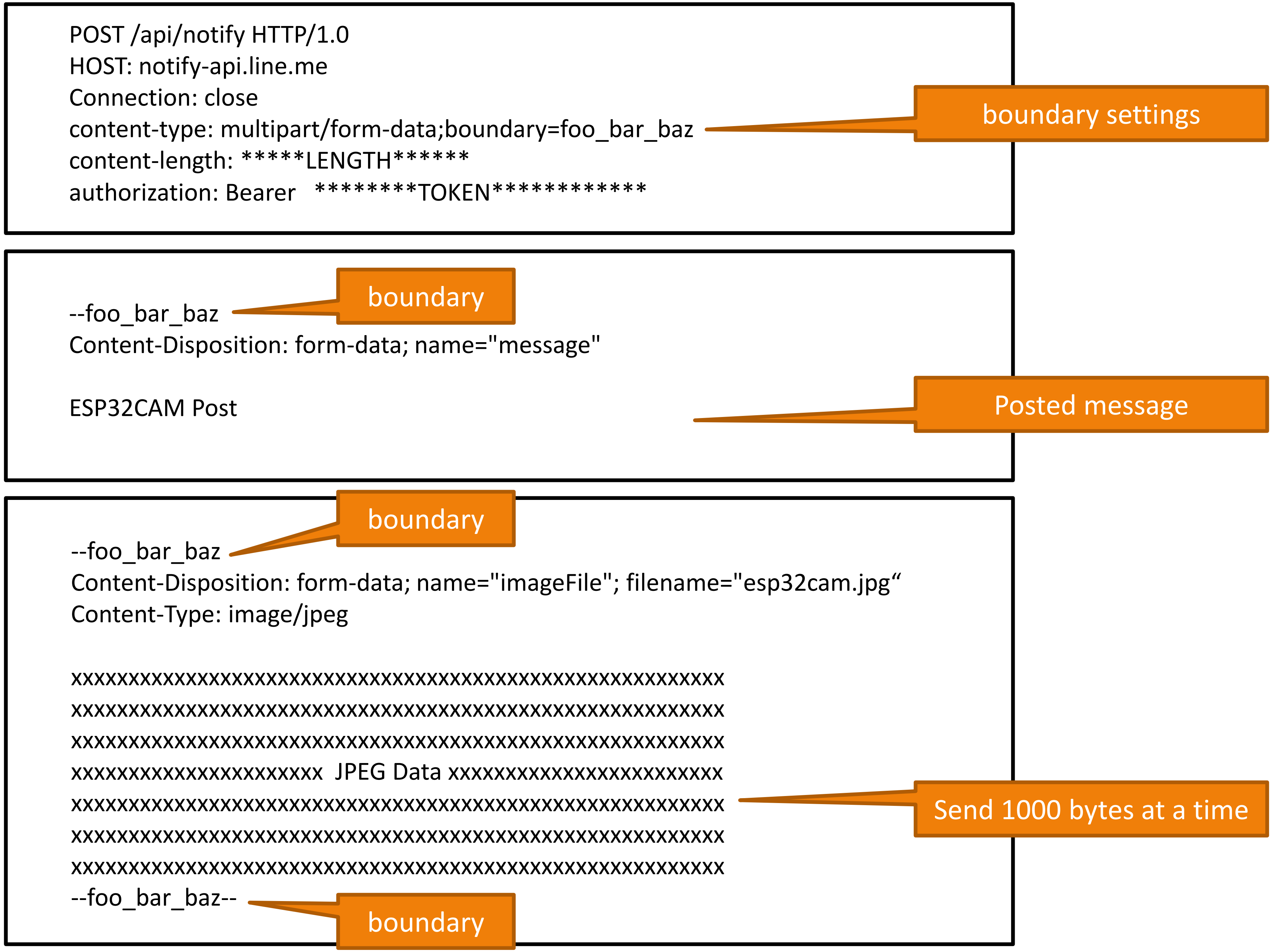
Processing when server connection fails

Server connection termination processing

JPEG image acquisition processing in the camera

5. Arduino program (send data)

Sent via HTTP(S) POST



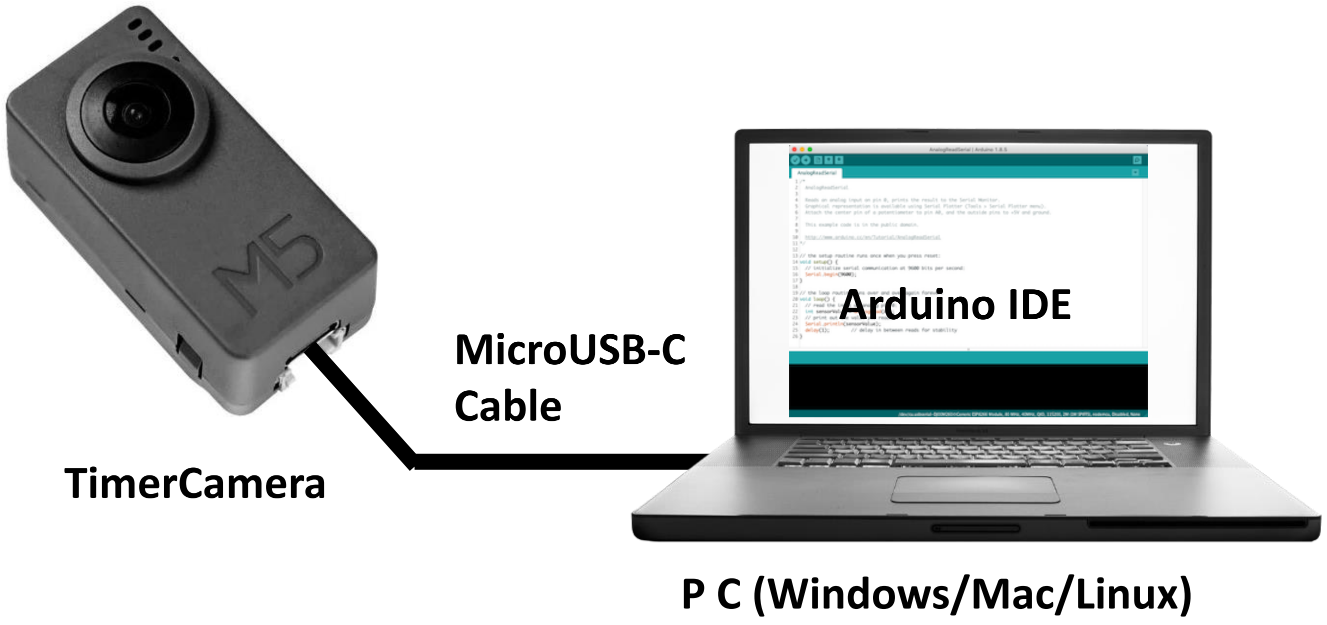
```
POST /api/notify HTTP/1.0
HOST: notify-api.line.me
Connection: close
content-type: multipart/form-data;boundary=foo_bar_baz
content-length: *****LENGTH******
authorization: Bearer   ********TOKEN************
```

```

--foo_bar_baz
Content-Disposition: form-data; name="message"

ESP32CAM Post

```

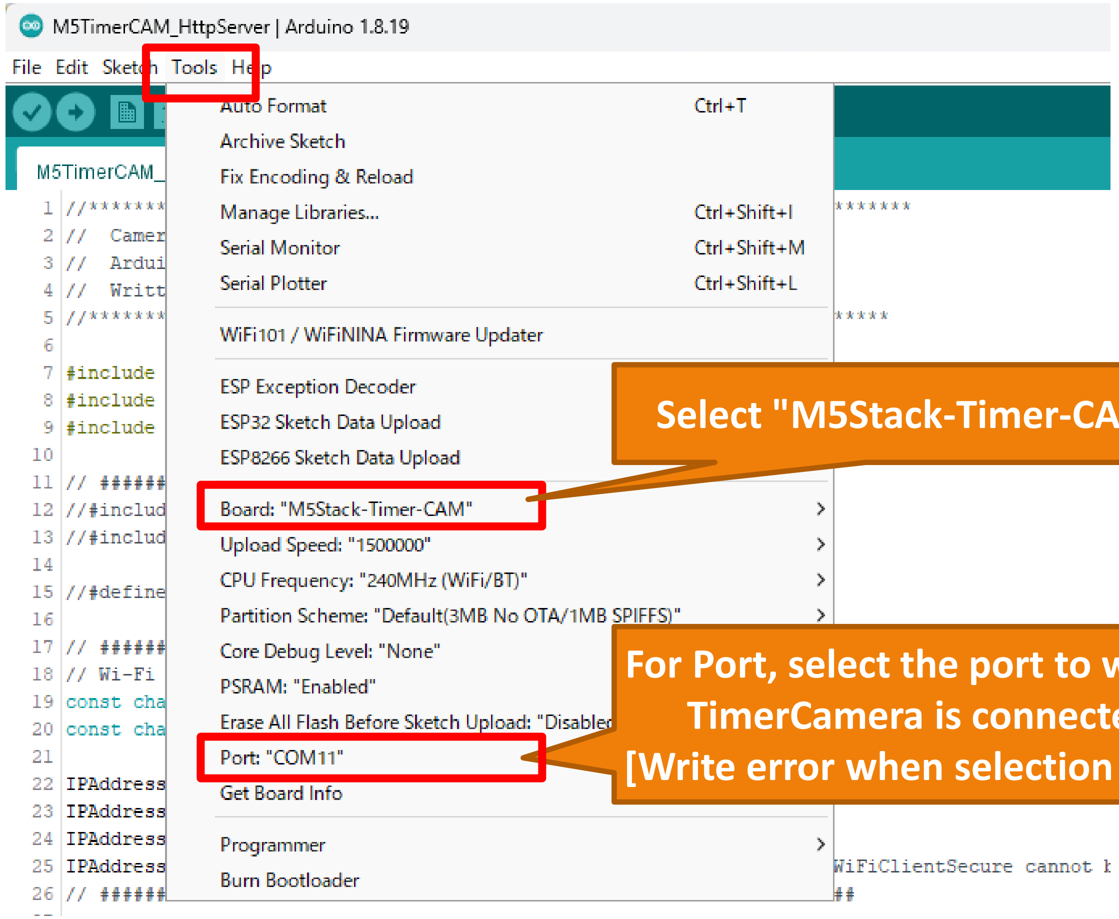
```

--foo_bar_baz
Content-Disposition: form-data; name="imageFile"; filename="esp32cam.jpg“
Content-Type: image/jpeg

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxx  JPEG Data xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
--foo_bar_baz--
```

6-1. Program writing

1) Connect TimeCamera with micro USB-C cable



2) Open the program with ArduinoIDE and check the settings again.
(Change the Wi-Fi settings [SSID, IP address, etc.] in the program.)


M5TimerCAM_HttpServer | Arduino 1.8.19
File Edit Sketch Tools Help
M5TimerCAM_
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
ESP Exception Decoder
ESP32 Sketch Data Upload
ESP8266 Sketch Data Upload
Get Board Info
Programmer
Burn Bootloader

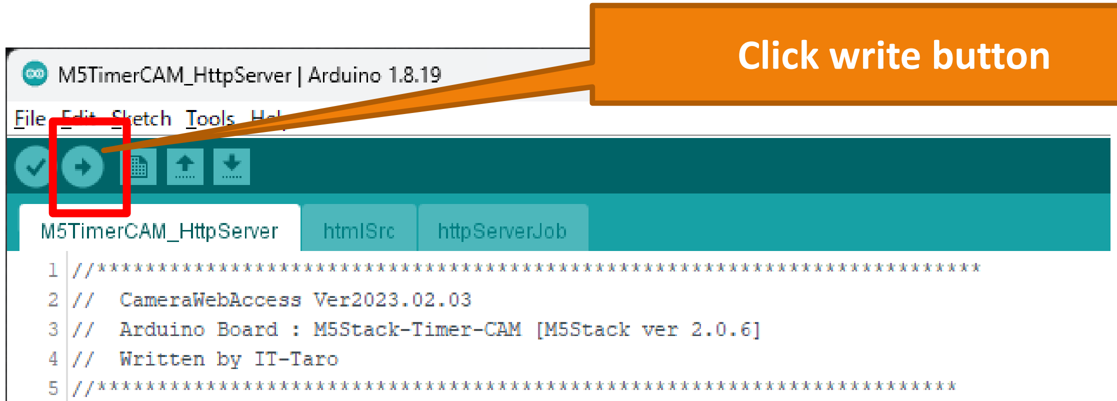
Board: "M5Stack-Timer-CAM"
Upload Speed: "1500000"
CPU Frequency: "240MHz (WiFi/BT)"
Partition Scheme: "Default(3MB No OTA/1MB SPIFFS)"
Core Debug Level: "None"
PSRAM: "Enabled"
Erase All Flash Before Sketch Upload: "Disabled
Port: "COM11"

```
//*******                    *******
//  Camer
//  Ardui
//  Writt
//*******                    *****

#include
#include
#include

// ######
//#includ
//#includ

//#define

// ######
// Wi-Fi
const cha
const cha

IPAddress
IPAddress
IPAddress
IPAddress                    WiFiClientSecure cannot k
// ######                    ##
```

Select "M5Stack-Timer-CAM"

For Port, select the port to which TimerCamera is connected [Write error when selection fails]

3) Click write button

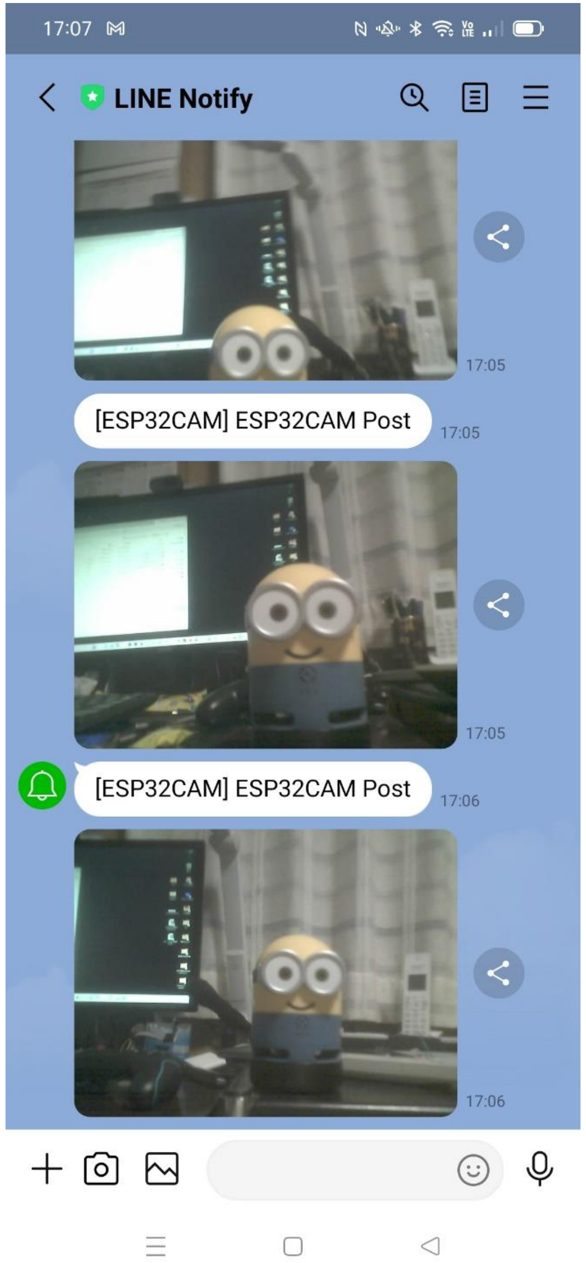

M5TimerCAM_HttpServer | Arduino 1.8.19
File Edit Sketch Tools Help
M5TimerCAM_HttpServer
htmlSrc
httpServerJob

```
//***************************************************************************
//  CameraWebAccess Ver2023.02.03
//  Arduino Board : M5Stack-Timer-CAM [M5Stack ver 2.0.6]
//  Written by IT-Taro
//***************************************************************************
```

Click write button

6-2. Operation check

When TimerCamera starts up, it acquires an image and posts it to LINE.



17:05

[ESP32CAM] ESP32CAM Post 17:05

17:05

[ESP32CAM] ESP32CAM Post 17:06

17:06

Posted at the time before the last boot

Posted on last boot

Messages and images posted to LINE by TimerCamera