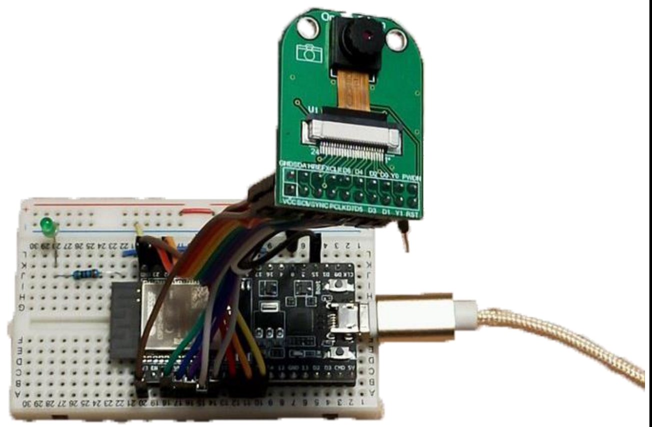
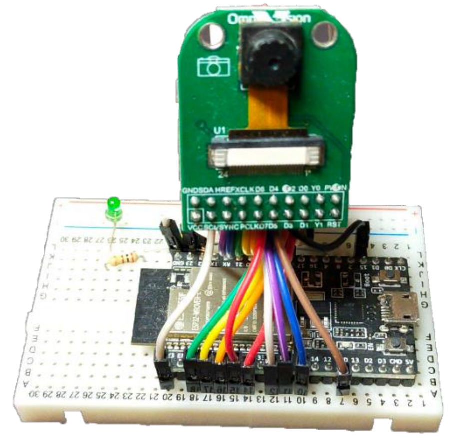
# スマホで動画視聴
# （M5Stack TimerCamera）

TimerCamera(ESP32)によるWebサーバ
及び、配信サーバの実装

# 目　次　《スマホで動画視聴》

## 1ー1. 機器選定（4000円以下程度を目標）

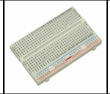
※費用は時期により変動しますので参考です。

| | 同じハードウェア構成 | | 近いハードウェア構成 | |
|---|---|---|---|---|
| | ①ESP32(WROOM)とOV2640 | ②M5Stack UnitCam (OV2640) | ③ESP32(WROVER)とOV2640 | ④M5Stack TimerCamera (OV3660) |
| 図 |  | |  | |
| 仕様 | メモリ[SRAM]: 520kbyte、解像度: 2M pixel | | メモリ[SRAM]: 8Mbyte | |
| | | プログラム書込にはキットが必要*1 | 解像度: 2M pixel | 解像度: 3M pixel |
| 用途 | 静止画 | | 静止画、動画 | |
| 費用 | 3930円 | M5Stack: UnitCam 18.95USD [marutsu: 2946円] + 1100円*1 | 4080円 | M5Stack: F)19.95, X)17.95USD [SwitchSience: F)2860, X)2596円] |
| ソフト | ほぼ流用可能 （Arduinoのマザーボード設定やポートの使い方に違いがある） | | | |
| 投稿 | GoogleAPI, GoogleAppScript[GAS]による画像のGoogleDrive保存 | - | - | スマホで動画視聴 今回 |

# １－２．ESP32でのカメラ利用（価格詳細）

※費用は時期により変動しますので参考です。
※Hobby-ITサイトからExcelダウンロード可能

## ①ESP32(WROOM)とOV2640　【3930円】

| NO | 項目 | 数量 | イメージ | 商品名 | URL | 購入先 | 価格 | 備考 |
|---|---|---|---|---|---|---|---|---|
| 1 | ESP32開発ボード WROOM | 1 | | ＥＳＰ３２－ＤｅｖＫｉｔＣ－３２Ｅ　ＥＳＰ３２－ＷＲＯＯＭ－３２Ｅ開発ボード　４ＭＢ | https://akizukidenshi.com/catalog/g/gM-15673/ | 秋月電子 | 1600 | 19Pin×2列仕様（他社は15Pin×2列） |
| 2 | ブレッドボード ６穴版　ＥＩＣ－３９０１ | 1 |  | ブレッドボード　６穴版　ＥＩＣ－３９０１ | https://akizukidenshi.com/catalog/g/gP-12366/ | | 460 | |
| 3 | 緑LED | 1 | | ３ｍｍ黄緑色ＬＥＤ　５７０ｎｍ７０度　ＯＳＧ８ＨＡ３Ｚ７４Ａ | https://akizukidenshi.com/catalog/g/gI-11637/ | | 10 | 状態表示用 |
| 4 | ＯＶ２６４０ カメラモジュール | 1 | | ＯＶ２６４０使用２００万画素カメラ　Ｂ００１１ | https://akizukidenshi.com/catalog/g/gM-13197/ | | 1680 | |
| 5 | ジャンパー ケーブル | 1 | | コネクタ付ケーブル　２０ｃｍ４０Ｐ　オスメス コネクタ付ケーブル　２０ｃｍ長 | https://akizukidenshi.com/catalog/g/gC-17228/ | | 180 | 今回は手持ちを利用したのでコネクタ形状など未確認 |
| 総合計 | | | | | | | **3,930** | **別途送料が必要です** |

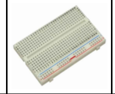
配線用のジャンパー線セットやLED抵抗は省略しました。

## ②M5Stack UnitCam　【4046円】

| NO | 項目 | 数量 | イメージ | 商品名 | URL | 購入先 | 価格 | 備考 | |
|---|---|---|---|---|---|---|---|---|---|
| 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シリアル変換モジュール | 1 | | ＦＴ２３４Ｘ　超小型ＵＳＢシリアル変換モジュール | https://akizukidenshi.com/catalog/g/gM-08461/ | 秋月電子 | 680 | UnitCam多数利用でも一つあれば良いです | ソフトウェアの書込に必要 |
| 3 | 細ピンヘッダ １×２０ | 1 | | 細ピンヘッダ　１×２０ | https://akizukidenshi.com/catalog/g/gC-04398/ | | 20 | | |
| 4 | ジャンパー ケーブル | 1 | | コネクタ付ケーブル　２０ｃｍ４０Ｐ　オスメス コネクタ付ケーブル　２０ｃｍ長 | https://akizukidenshi.com/catalog/g/gC-17228/ | | 180 | 今回は手持ちを利用したのでコネクタ形状など未確認 | |
| 5 | ブレッドボード | 1 | | ブレッドボード　ＢＢ－８０１ | https://akizukidenshi.com/catalog/g/gP-05294/ | | 220 | | |
| 総合計 | | | | | | | **4,046** | **別途送料が必要です** | |

専用Uploaderもあるが、汎用性があるので今回はこの物品を選択

## ③ESP32(WROVER)とOV2640　【4080円】

| NO | 項目 | 数量 | イメージ | 商品名 | URL | 購入先 | 価格 | 備考 |
|---|---|---|---|---|---|---|---|---|
| 1 | ESP32開発ボード WROVER | 1 | | ＥＳＰ３２－ＤｅｖＫｉｔＣ－ＶＥ　ＥＳＰ３２－ＷＲＯＶＥＲ－Ｅ開発ボード　８ＭＢ | https://akizukidenshi.com/catalog/g/gM-15674/ | 秋月電子 | 1750 | 19Pin×2列仕様（他社は15Pin×2列） |
| 2 | ブレッドボード ６穴版　ＥＩＣ－３９０１ | 1 |  | ブレッドボード　６穴版　ＥＩＣ－３９０１ | https://akizukidenshi.com/catalog/g/gP-12366/ | | 460 | |
| 3 | 緑LED | 1 | | ３ｍｍ黄緑色ＬＥＤ　５７０ｎｍ７０度　ＯＳＧ８ＨＡ３Ｚ７４Ａ | https://akizukidenshi.com/catalog/g/gI-11637/ | | 10 | 状態表示用 |
| 4 | ＯＶ２６４０ カメラモジュール | 1 | | ＯＶ２６４０使用２００万画素カメラ　Ｂ００１１ | https://akizukidenshi.com/catalog/g/gM-13197/ | | 1680 | |
| 5 | ジャンパー ケーブル | 1 | | コネクタ付ケーブル　２０ｃｍ４０Ｐ　オスメス コネクタ付ケーブル　２０ｃｍ長 | https://akizukidenshi.com/catalog/g/gC-17228/ | | 180 | 今回は手持ちを利用したのでコネクタ形状など未確認 |
| 総合計 | | | | | | | **4,080** | **別途送料が必要です** |

配線用のジャンパー線セットやLED抵抗は省略しました。

## ④M5Stack TimerCamera(OV3660)　【2596/2860円】

| NO | 項目 | 数量 | イメージ | 商品名 | URL | 購入先 | 価格 | 備考 |
|---|---|---|---|---|---|---|---|---|
| 1 | Timer Camera X | 1 | | ESP32 PSRAM Timer Camera X (OV3660) | https://shop.m5stack.com/collections/m5-cameras | M5Stack | $17.95 | 視野角 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 | 視野角 120° |
| | | | | | https://www.switch-science.com/products/6786 | SWITCH SIENCE | 2860 | |
| 総合計 | | | | | | | **2,860** | **別途送料が必要です** |

X/Fは視野角の違い
マイクロＵＳＢケーブル付きでパソコンがあれば開発可能

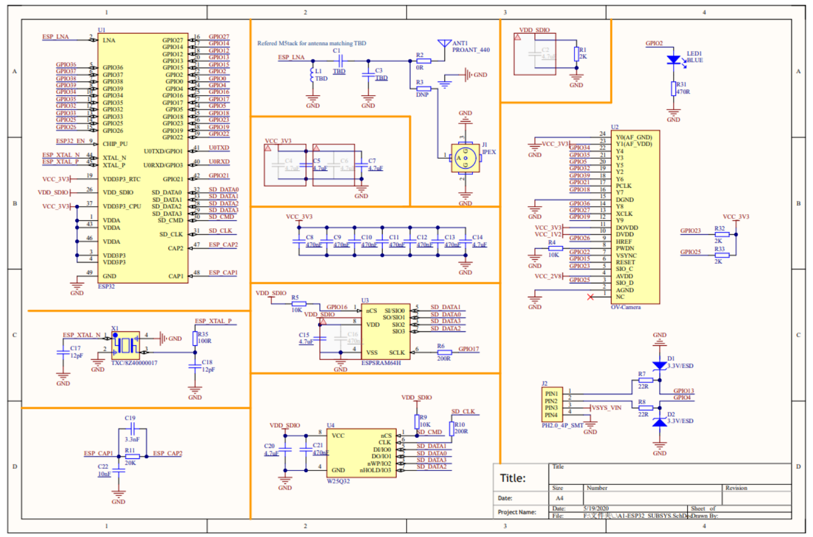
# 1－3. TimerCamera

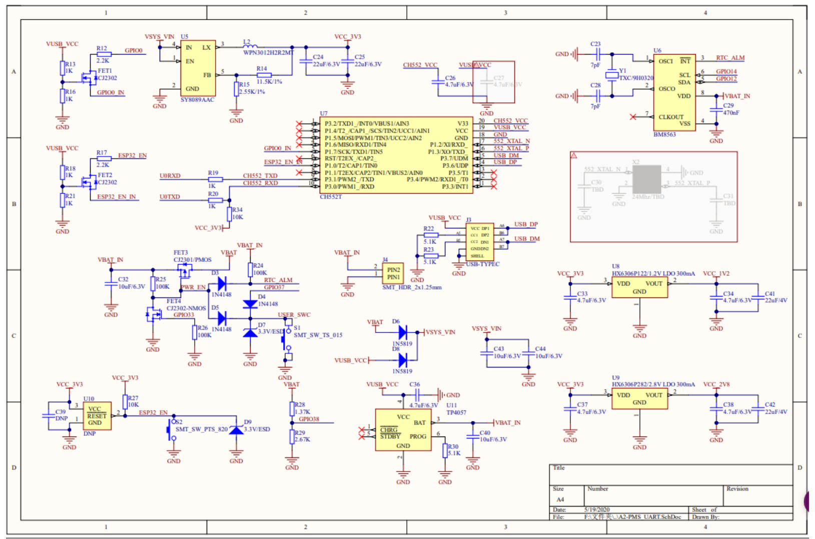
●M5Stack Official　TimerCamera Document
https://docs.m5stack.com/en/unit/timercam_x

● 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 |

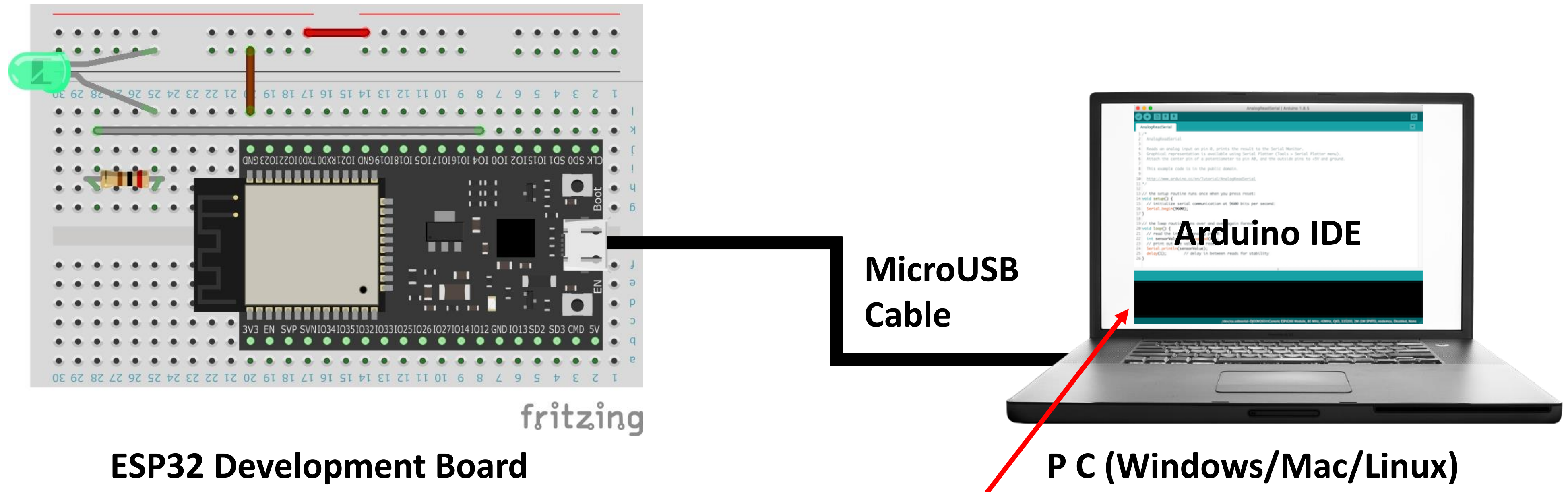
● Schematic





## 2. 開発環境

開発環境はArduinoを利用していきます。

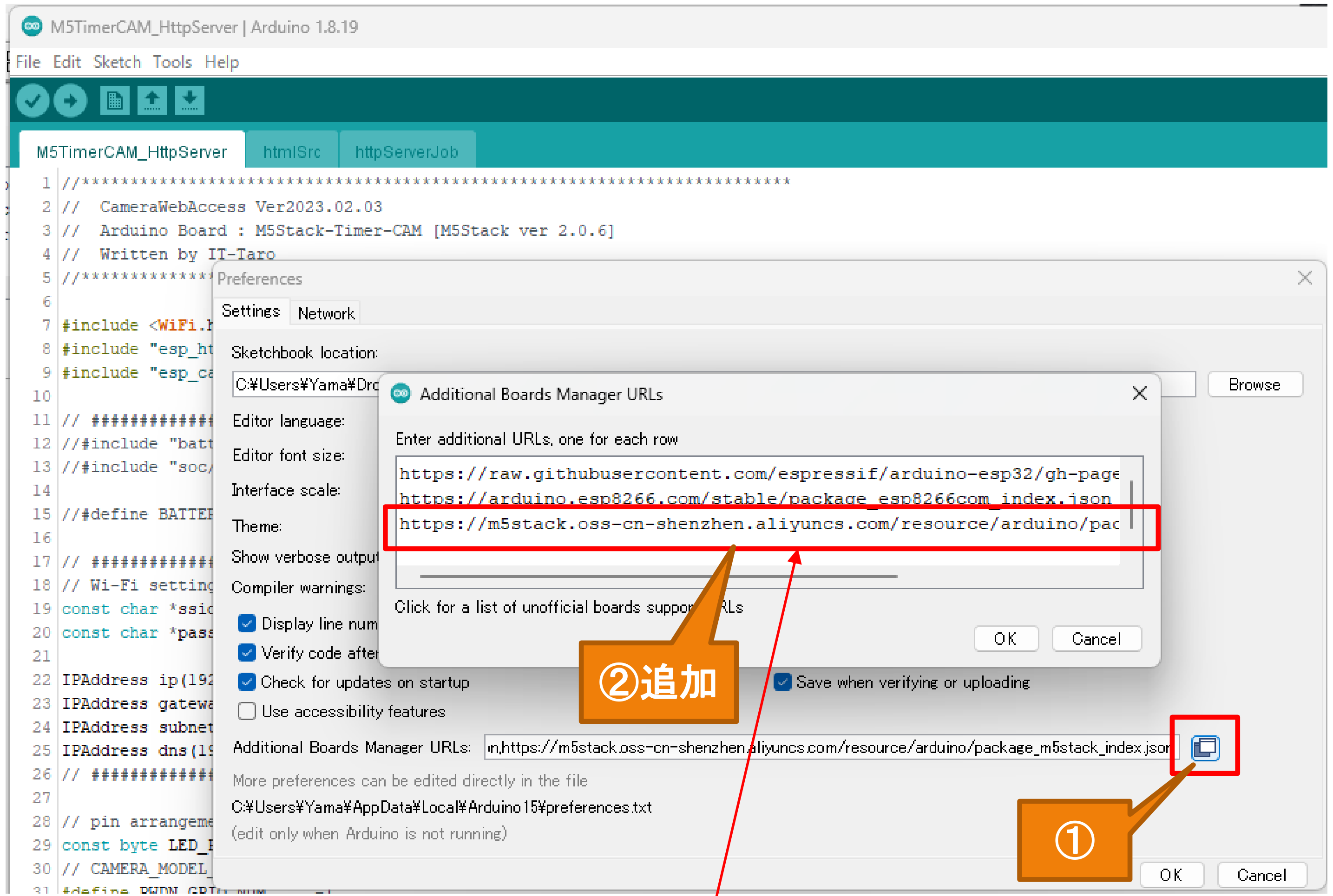


【Arduino Official site】
https://www.arduino.cc/
ダウンロード可能

# ３－１．Arduino設定（Board設定）

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

## 1) ArduinoIDE設定からAdditional Board Manager設定を追加



```
//*************************************************************************
//  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 BATTEF

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

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

// pin arrangeme
const byte LED_F
// CAMERA_MODEL_
```

Sketchbook location: C:¥Users¥Yama¥Drc
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)

Additional Boards Manager URLs
Enter additional URLs, one for each row

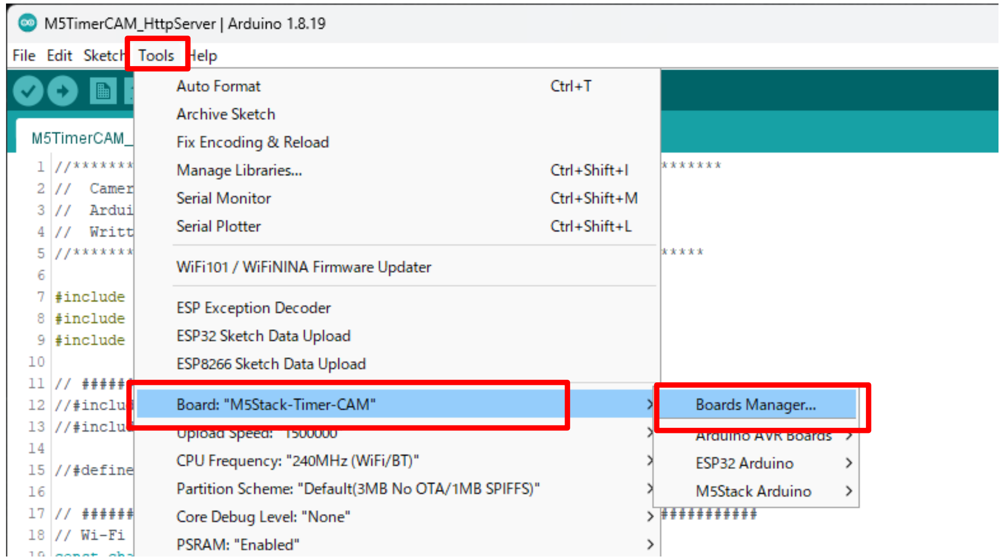
```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-page
https://arduino.esp8266.com/stable/package_esp8266com_index.json
https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/arduino/pac
```

Click for a list of unofficial boards suppor RLs

**設定値：**

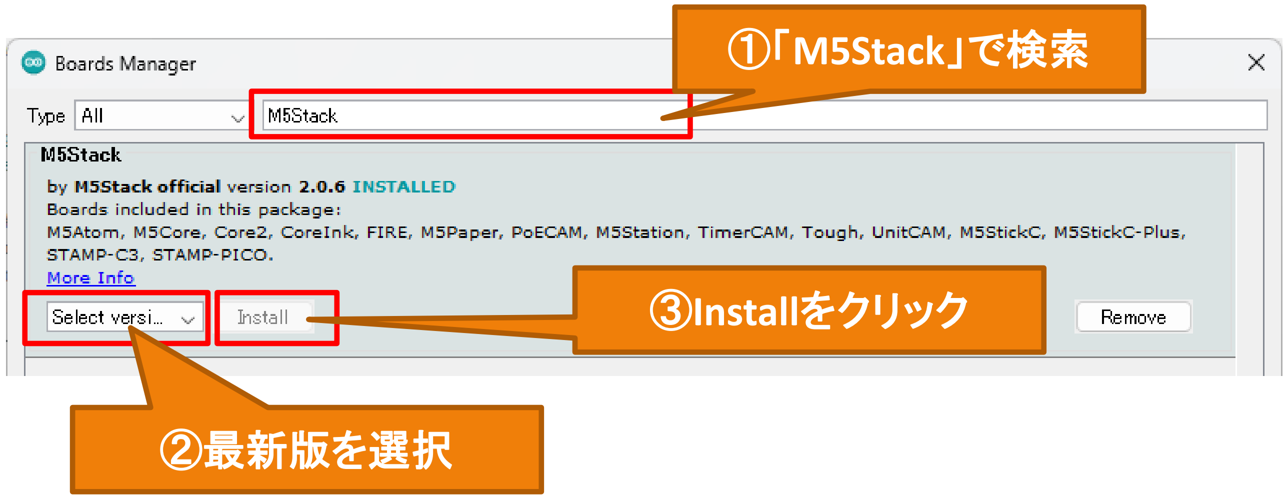
https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/arduino/package_m5stack_index.json

## 2) Board Managerを起動



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
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

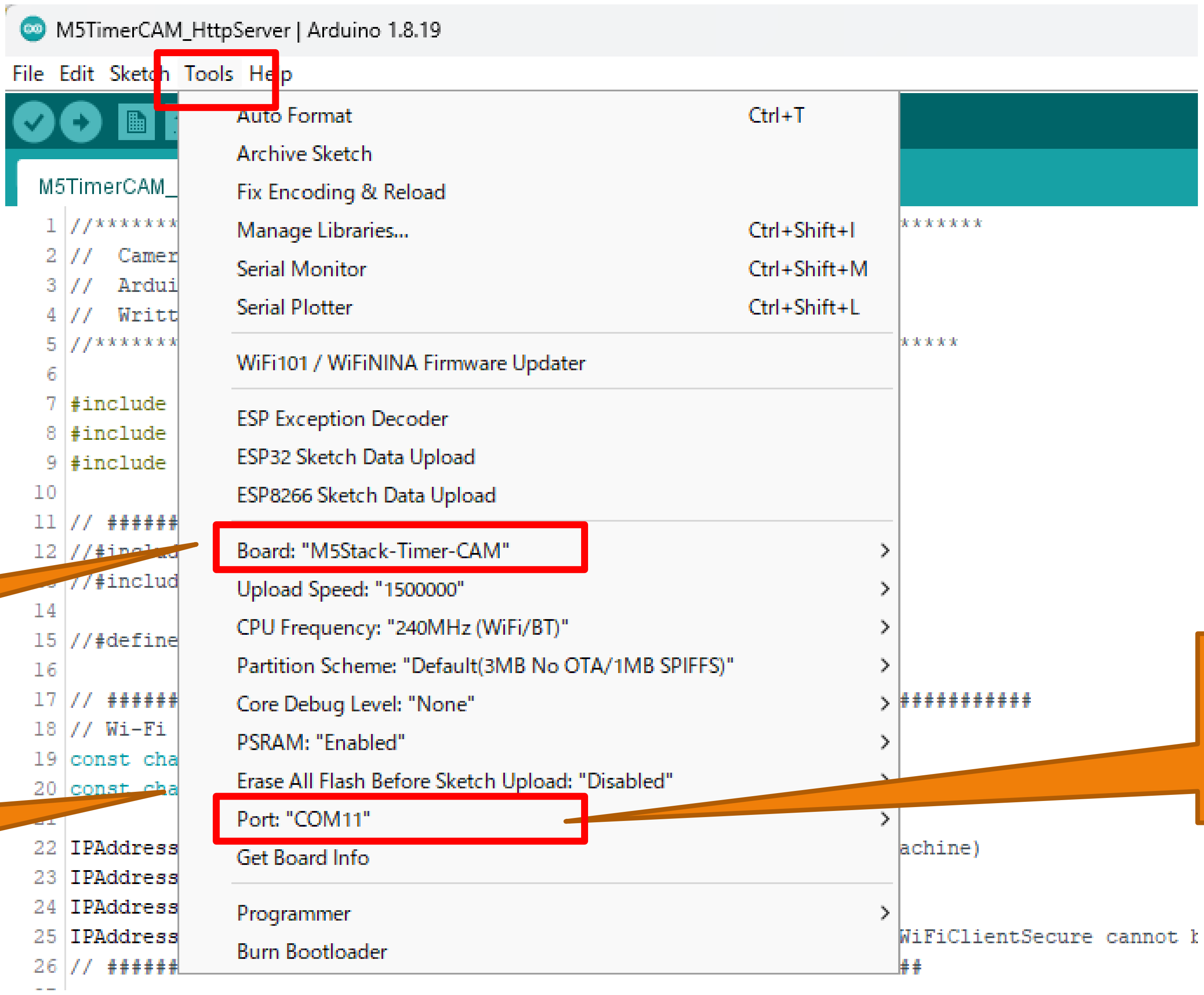
## 3) M5Stackをインストール



**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

# ３－１. Arduino設定（Board設定）

4) Boardを「M5Stack-Timer-CAM」に設定



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" | |
| Erase All Flash Before Sketch Upload: "Disabled" | |
| Port: "COM11" | |
| Get Board Info | |
| Programmer | |
| Burn Bootloader | |

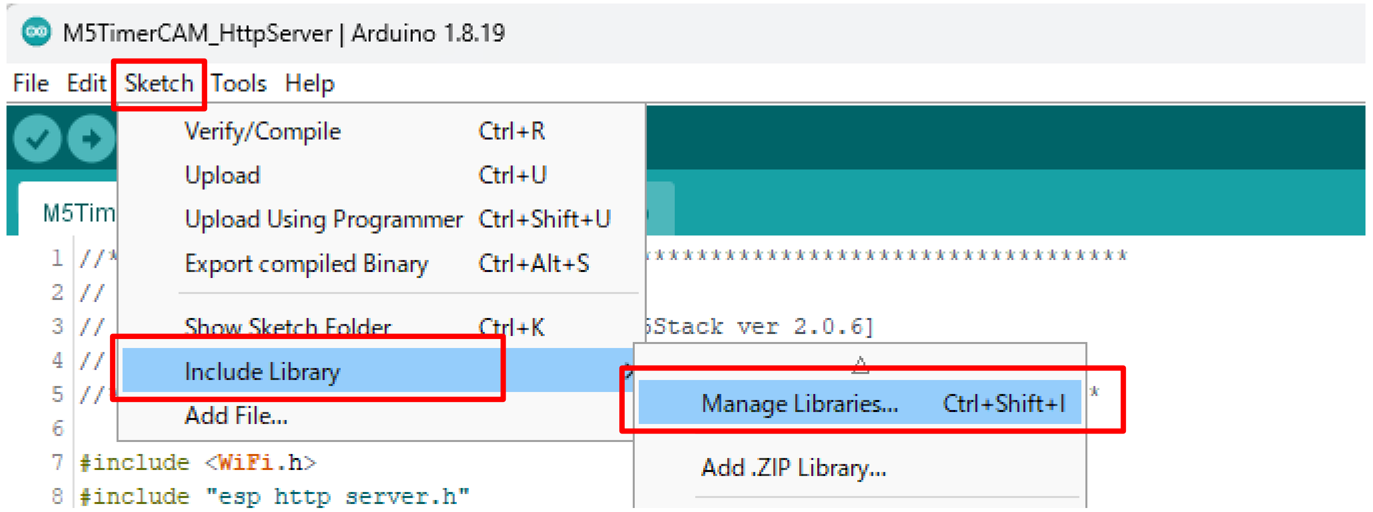
「M5Stack-Timer-CAM」を選択

他の設定は変更なし
（初期値のまま）

PortはTimerCameraが接続された
ポートを選択すること
【選択失敗時、書込エラー】

## ３－２．Arduino設定（Library追加）

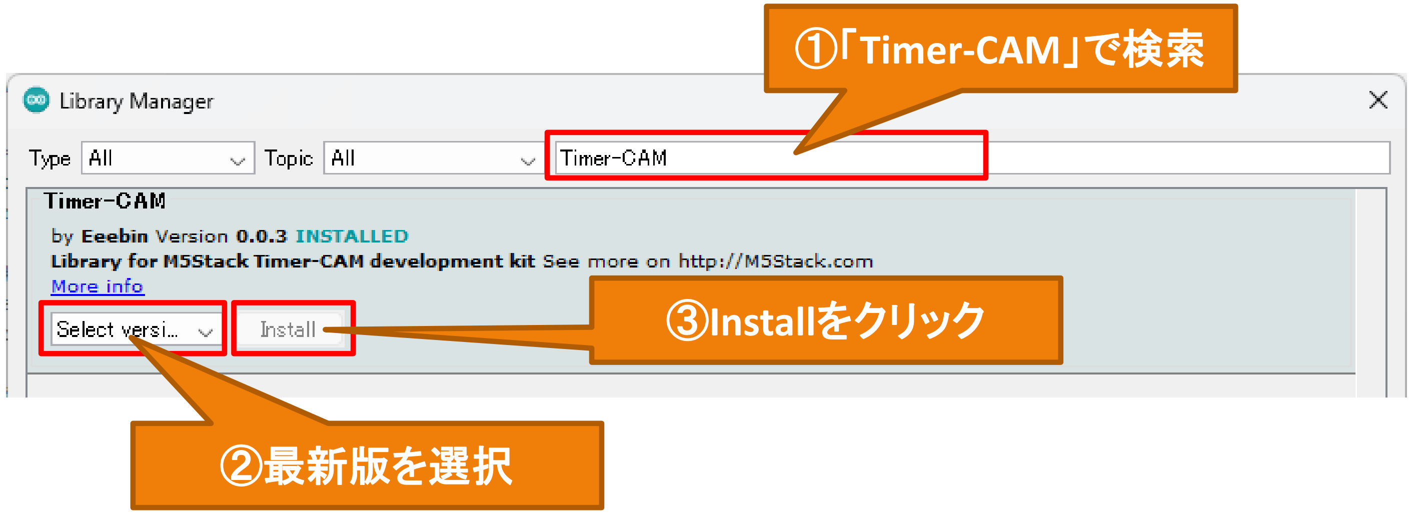
1) Library Managerを起動


M5TimerCAM_HttpServer | Arduino 1.8.19

File
Edit
Sketch
Tools
Help
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...

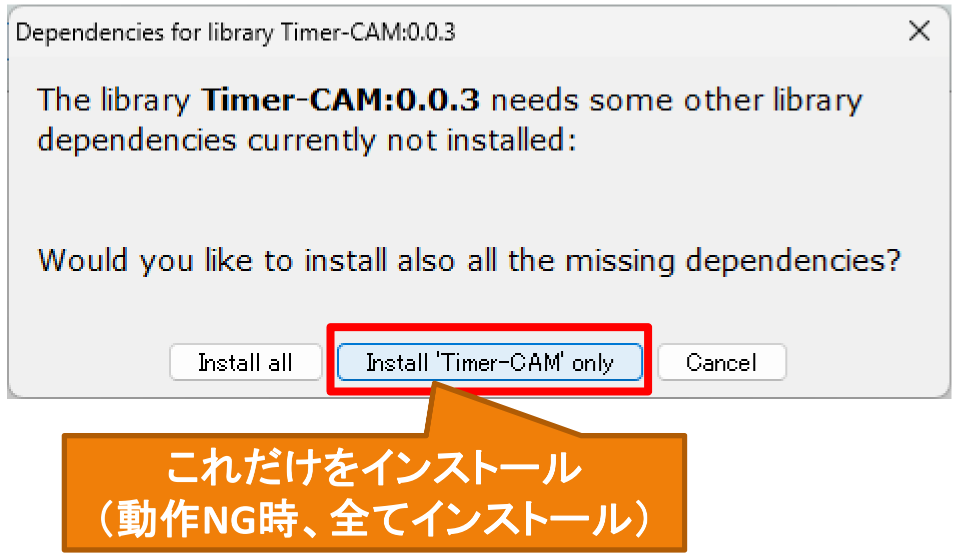
```
//*                                   ****************************************
//
//                                    5Stack ver 2.0.6]
//
//
#include <WiFi.h>
#include "esp http server.h"
```

2) 「Timer-CAM」をインストール



Timer-CAM
by Eeebin Version 0.0.3 INSTALLED
Library for M5Stack Timer-CAM development kit See more on http://M5Stack.com
More info

3) 「Timer-CAM」だけをインストール


Dependencies for library Timer-CAM:0.0.3

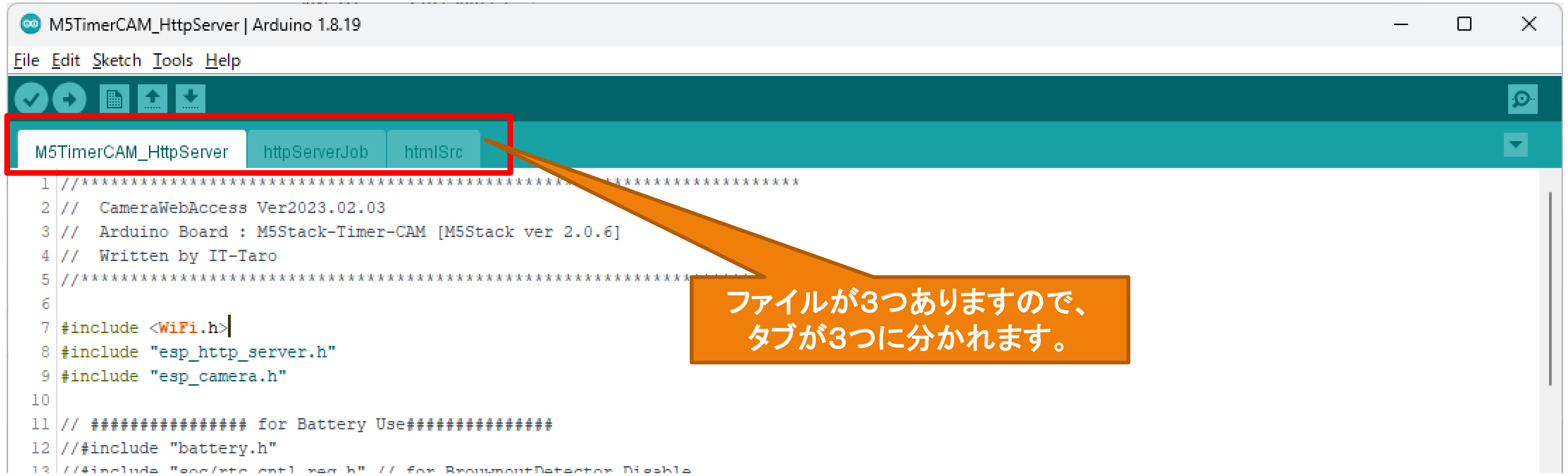
The library **Timer-CAM:0.0.3** needs some other library dependencies currently not installed:

Would you like to install also all the missing dependencies?

# ４ー１．Arduinoプログラム（ファイル構成）

## ●ファイル構成

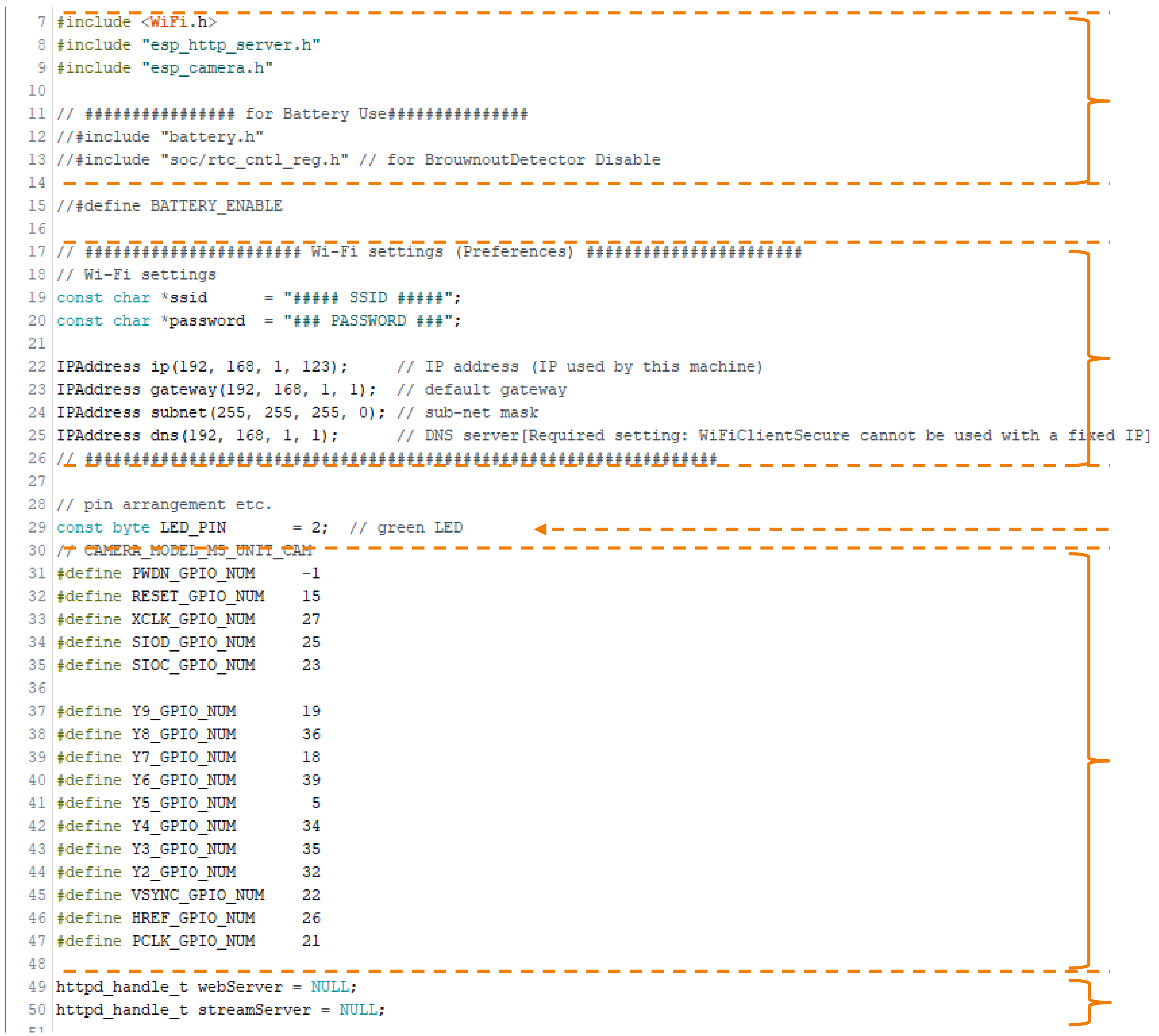
| | | | | プログラム種別 |
|---|---|---|---|---|
| M5TimerCAM_HttpServer | M5TimerCAM_HttpServer.ino | ： | setup関数やloop関数など | Arduino |
| | httpServerJob.ino | ： | Webサーバに関するプログラム | Arduino |
| | htmlSrc.ino | ： | 送信するHTMLファイル設定 | Arduino |


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

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

#include <WiFi.h>
#include "esp_http_server.h"
#include "esp_camera.h"

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

ファイルが3つありますので、
タブが3つに分かれます。

# ４－２. Arduinoプログラム（グローバル定義）



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

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

//#define BATTERY_ENABLE

// ####################### Wi-Fi settings (Preferences) #######################
// Wi-Fi settings
const char *ssid      = "##### SSID #####";
const char *password  = "### PASSWORD ###";

IPAddress ip(192, 168, 1, 123);     // IP address (IP used by this machine)
IPAddress gateway(192, 168, 1, 1);  // default gateway
IPAddress subnet(255, 255, 255, 0); // sub-net mask
IPAddress dns(192, 168, 1, 1);      // DNS server[Required setting: WiFiClientSecure cannot be used with a fixed IP]
// ###################################################################

// pin arrangement etc.
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

httpd_handle_t webServer = NULL;
httpd_handle_t streamServer = NULL;
```

ライブラリの読み込み

Wi-Fi設定

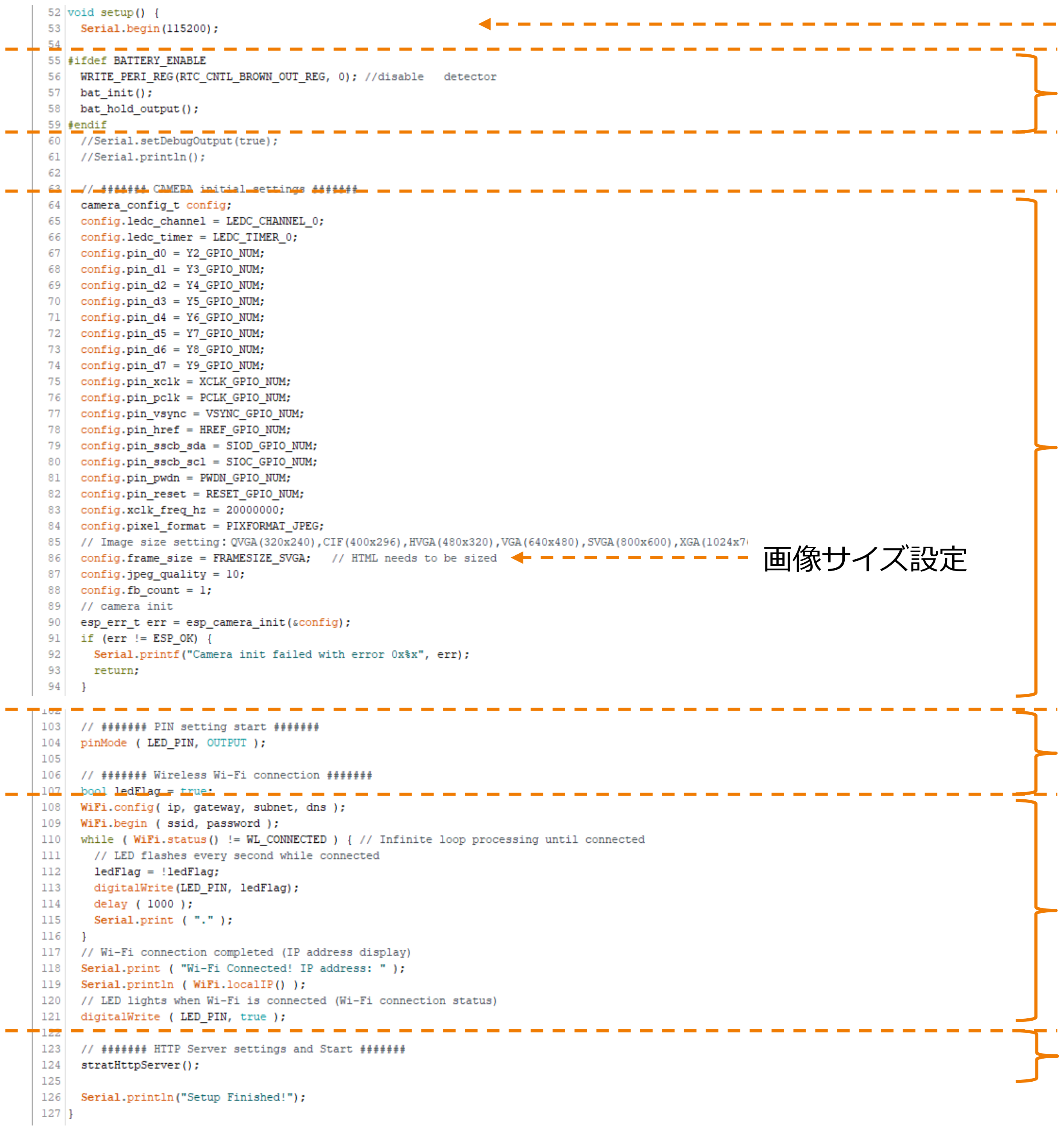
**設定変更必要**

LED　ポート設定

カメラ　ポート設定

Webサーバ及び、配信サーバを定義

# ４－３．Arduinoプログラム（Setup関数）



```
void setup() {
  Serial.begin(115200);

#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 initial settings #######
  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(1024x7
  config.frame_size = FRAMESIZE_SVGA;   // HTML needs to be sized
  config.jpeg_quality = 10;
  config.fb_count = 1;
  // 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;
  }

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

  // ####### Wireless Wi-Fi connection #######
  bool ledFlag = true;
  WiFi.config( ip, gateway, subnet, dns );
  WiFi.begin ( ssid, password );
  while ( WiFi.status() != WL_CONNECTED ) { // Infinite loop processing 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() );
  // LED lights when Wi-Fi is connected (Wi-Fi connection status)
  digitalWrite ( LED_PIN, true );

  // ####### HTTP Server settings and Start #######
  stratHttpServer();

  Serial.println("Setup Finished!");
}
```

シリアルモニタ開始

バッテリー利用時、BrownOUT（電圧低下）チェックをOFF
［すぐにエラーになり再起動してしまう場合に有効設定へ］

カメラ　ポート設定及び初期化

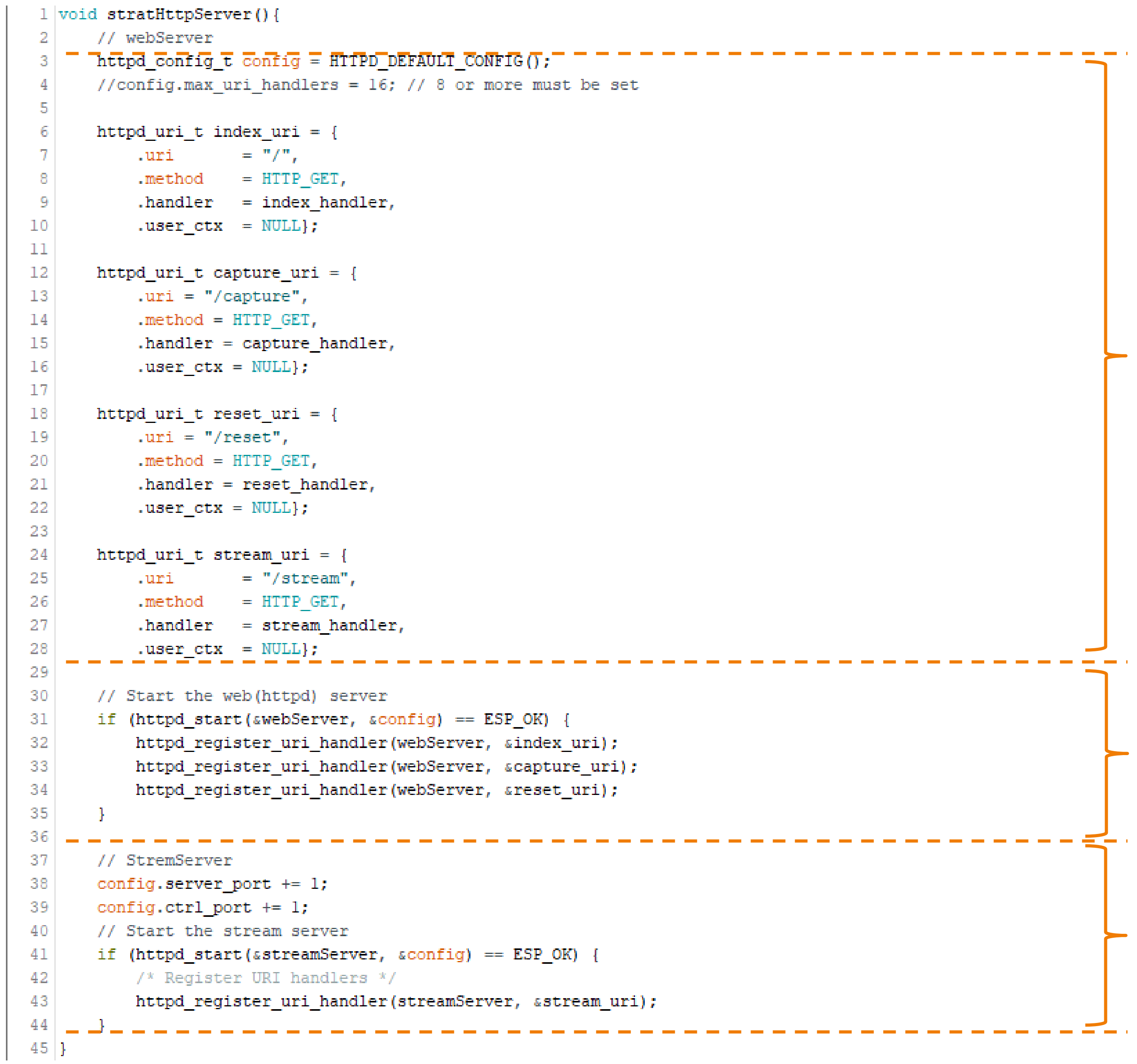
画像サイズ設定

LED　ポート設定し、点灯へ

Wi-Fi設定し、接続へ

Webサーバ起動

# ４－４．Arduinoプログラム（stratHttpServer関数）



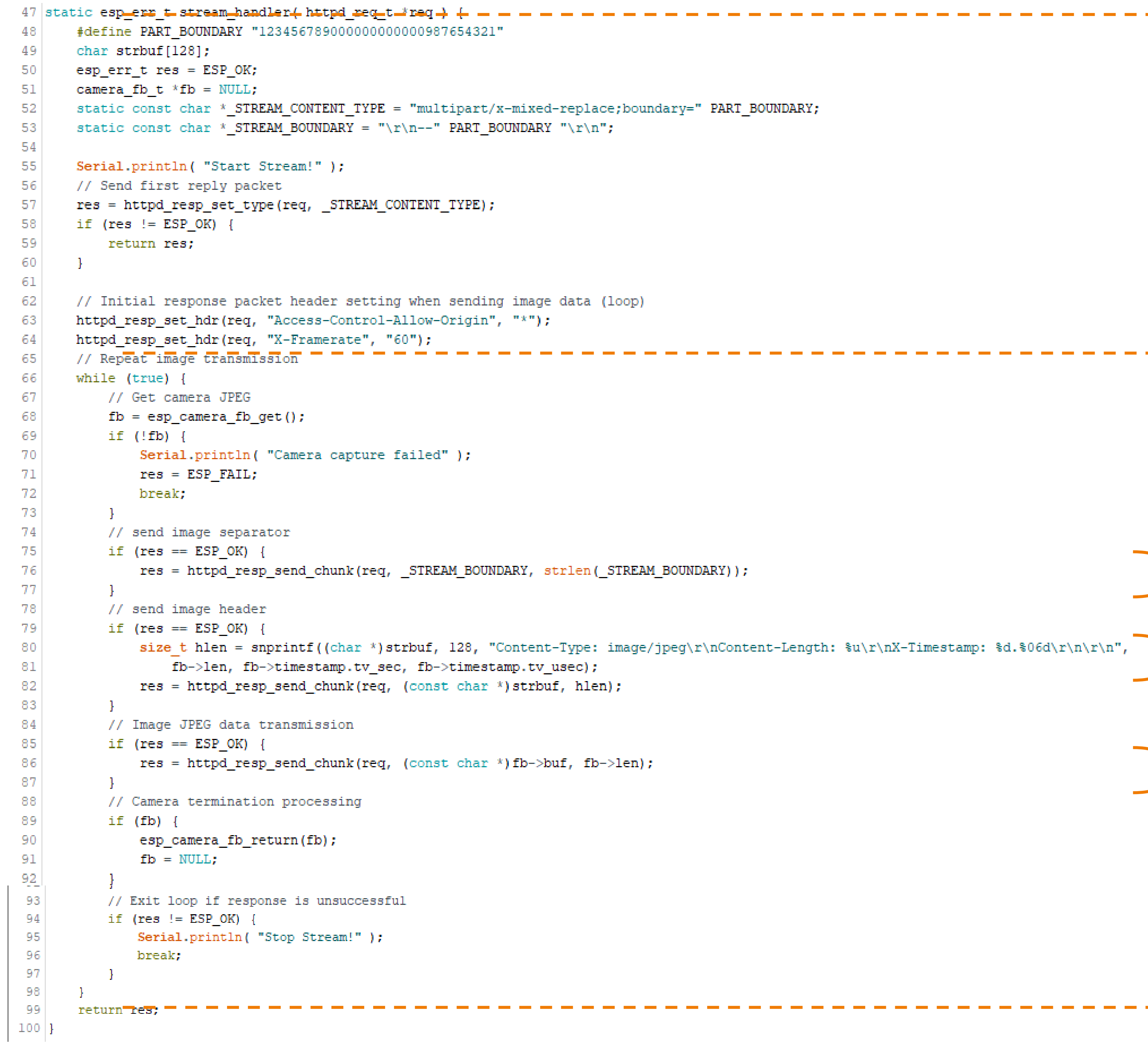
```
void stratHttpServer(){
    // webServer
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    //config.max_uri_handlers = 16; // 8 or more must be set

    httpd_uri_t index_uri = {
        .uri        = "/",
        .method     = HTTP_GET,
        .handler    = index_handler,
        .user_ctx   = NULL};

    httpd_uri_t capture_uri = {
        .uri = "/capture",
        .method = HTTP_GET,
        .handler = capture_handler,
        .user_ctx = NULL};

    httpd_uri_t reset_uri = {
        .uri = "/reset",
        .method = HTTP_GET,
        .handler = reset_handler,
        .user_ctx = NULL};

    httpd_uri_t stream_uri = {
        .uri        = "/stream",
        .method     = HTTP_GET,
        .handler    = stream_handler,
        .user_ctx   = NULL};

    // Start the web(httpd) server
    if (httpd_start(&webServer, &config) == ESP_OK) {
        httpd_register_uri_handler(webServer, &index_uri);
        httpd_register_uri_handler(webServer, &capture_uri);
        httpd_register_uri_handler(webServer, &reset_uri);
    }

    // StremServer
    config.server_port += 1;
    config.ctrl_port += 1;
    // Start the stream server
    if (httpd_start(&streamServer, &config) == ESP_OK) {
        /* Register URI handlers */
        httpd_register_uri_handler(streamServer, &stream_uri);
    }
}
```

アクセスURLと処理を定義

Webサーバを起動
（ポート番号：80）

配信サーバを起動
（ポート番号：81）

# ４ー５. Arduinoプログラム（stream_handler関数）



```
static esp_err_t stream_handler( httpd_req_t *req ) {
    #define PART_BOUNDARY "123456789000000000000987654321"
    char strbuf[128];
    esp_err_t res = ESP_OK;
    camera_fb_t *fb = NULL;
    static const char *_STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary=" PART_BOUNDARY;
    static const char *_STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";

    Serial.println( "Start Stream!" );
    // Send first reply packet
    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
    if (res != ESP_OK) {
        return res;
    }

    // Initial response packet header setting when sending image data (loop)
    httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
    httpd_resp_set_hdr(req, "X-Framerate", "60");
    // Repeat image transmission
    while (true) {
        // Get camera JPEG
        fb = esp_camera_fb_get();
        if (!fb) {
            Serial.println( "Camera capture failed" );
            res = ESP_FAIL;
            break;
        }
        // send image separator
        if (res == ESP_OK) {
            res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY, strlen(_STREAM_BOUNDARY));
        }
        // send image header
        if (res == ESP_OK) {
            size_t hlen = snprintf((char *)strbuf, 128, "Content-Type: image/jpeg\r\nContent-Length: %u\r\nX-Timestamp: %d.%06d\r\n\r\n",
                fb->len, fb->timestamp.tv_sec, fb->timestamp.tv_usec);
            res = httpd_resp_send_chunk(req, (const char *)strbuf, hlen);
        }
        // Image JPEG data transmission
        if (res == ESP_OK) {
            res = httpd_resp_send_chunk(req, (const char *)fb->buf, fb->len);
        }
        // Camera termination processing
        if (fb) {
            esp_camera_fb_return(fb);
            fb = NULL;
        }
        // Exit loop if response is unsuccessful
        if (res != ESP_OK) {
            Serial.println( "Stop Stream!" );
            break;
        }
    }
    return res;
}
```

送信ヘッダ内容を定義

各画像データの
区切りを送信

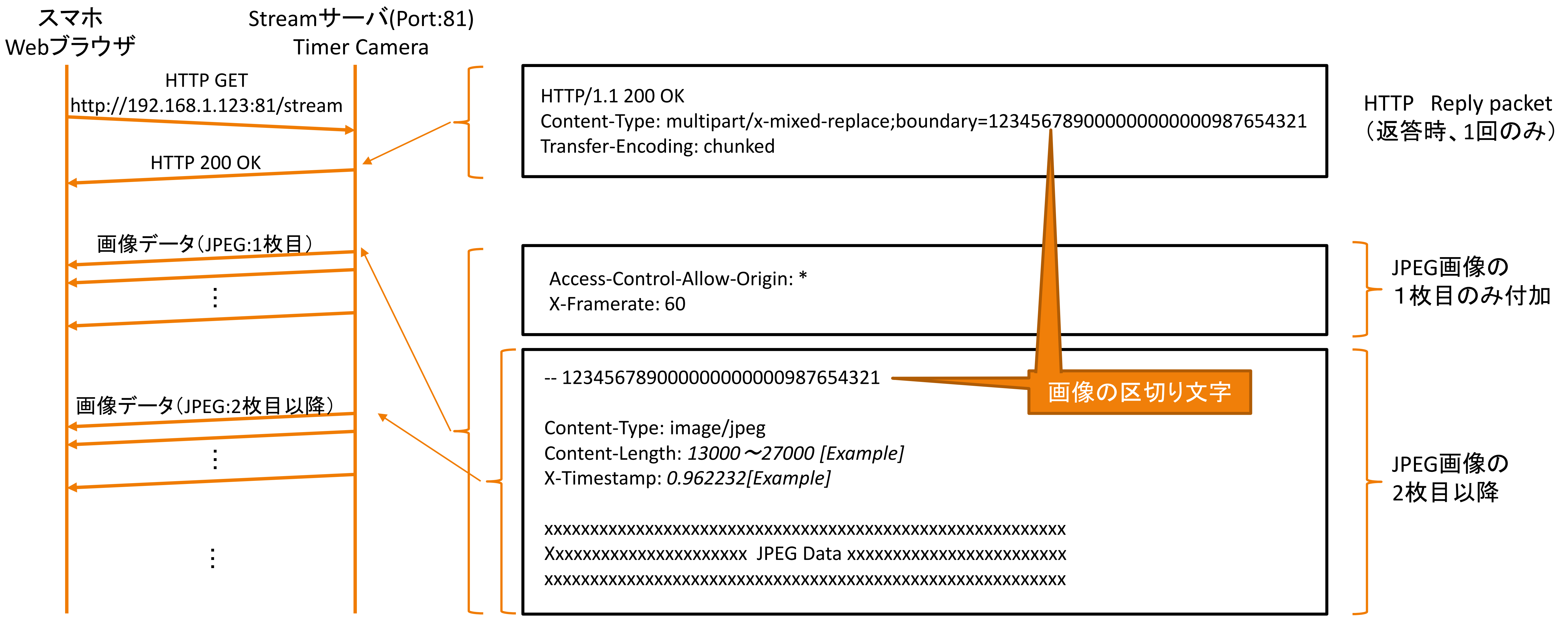
送信データの
ヘッダ情報送信

画像データを送信

Jpeg画像を取得し、端末へ送信
（1画像づつ取得し、送信）

# ４－５．Arduinoプログラム（送信データ）

● 動画配信の主な通信（Motion JPEG）
TCPのAck（応答確認）のような通信は省略しています。

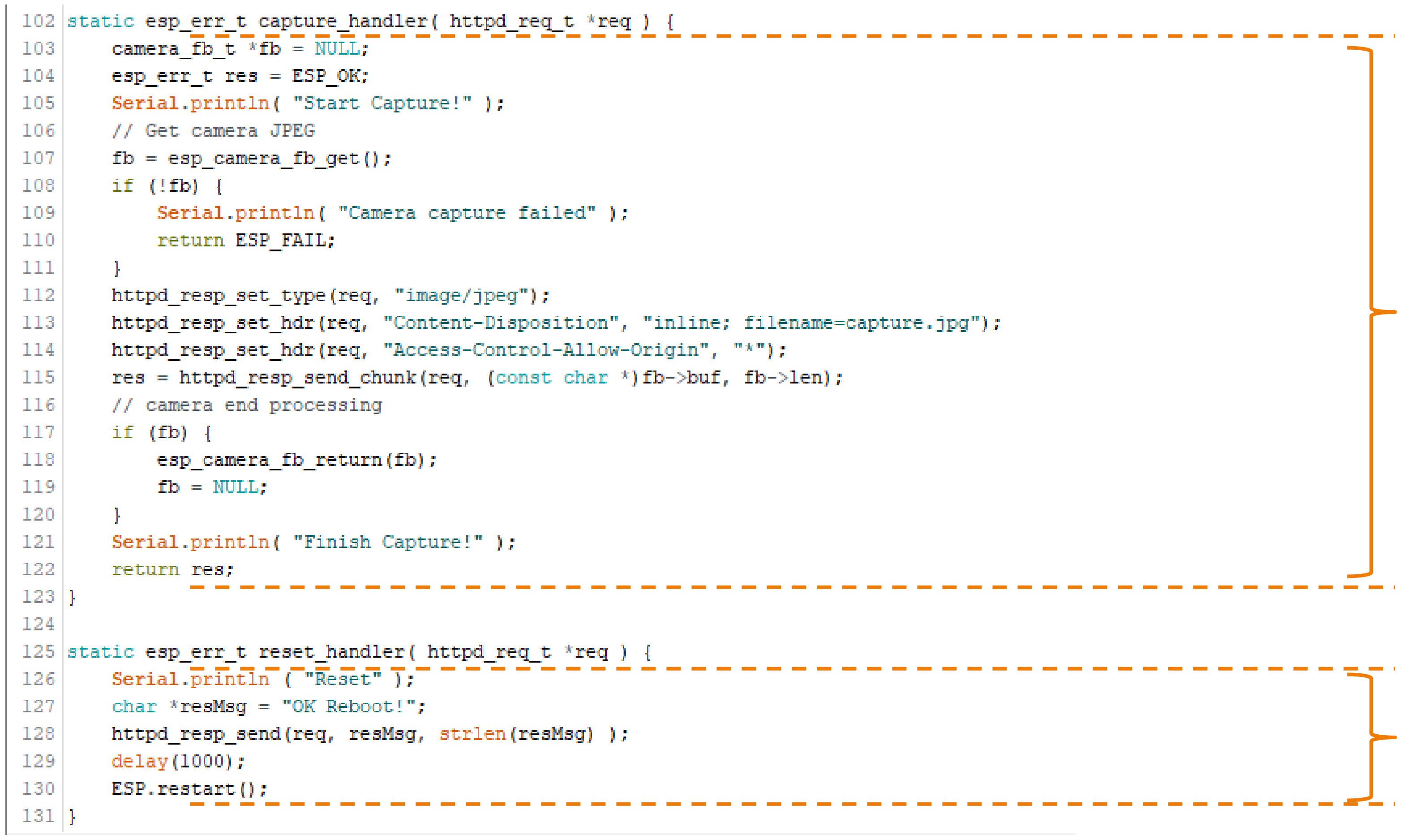


```
HTTP/1.1 200 OK
Content-Type: multipart/x-mixed-replace;boundary=123456789000000000000987654321
Transfer-Encoding: chunked
```

```
Access-Control-Allow-Origin: *
X-Framerate: 60
```

```
-- 123456789000000000000987654321

Content-Type: image/jpeg
Content-Length: 13000～27000 [Example]
X-Timestamp: 0.962232[Example]

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Xxxxxxxxxxxxxxxxxxxxxx  JPEG Data xxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

# ４－６. Arduinoプログラム（capture_handler, reset_handler関数）



```
static esp_err_t capture_handler( httpd_req_t *req ) {
    camera_fb_t *fb = NULL;
    esp_err_t res = ESP_OK;
    Serial.println( "Start Capture!" );
    // Get camera JPEG
    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println( "Camera capture failed" );
        return ESP_FAIL;
    }
    httpd_resp_set_type(req, "image/jpeg");
    httpd_resp_set_hdr(req, "Content-Disposition", "inline; filename=capture.jpg");
    httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
    res = httpd_resp_send_chunk(req, (const char *)fb->buf, fb->len);
    // camera end processing
    if (fb) {
        esp_camera_fb_return(fb);
        fb = NULL;
    }
    Serial.println( "Finish Capture!" );
    return res;
}

static esp_err_t reset_handler( httpd_req_t *req ) {
    Serial.println ( "Reset" );
    char *resMsg = "OK Reboot!";
    httpd_resp_send(req, resMsg, strlen(resMsg) );
    delay(1000);
    ESP.restart();
}
```

ONE-SHOT実行時、画像を取得し送信
（画像1枚だけ処理）

REBOOT実行時、ESP32を再起動

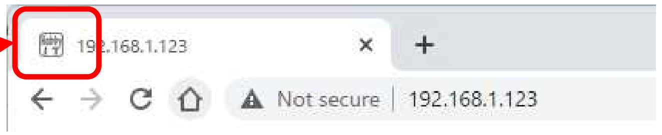
「htmlSrc.ino」ファイルはHTMLを変数定義しているだけですので、HTMLプログラムで理解していきます。

# 5. HTMLプログラム

```html
<!DOCTYPE html><html lang="jp"><head><meta charset="UTF-8"/>
<style type="text/css"><!--
#contents { max-width: 800px; }
img { width:100%;height:600px; background-color:grey; }
h1 { margin: 0px; font-size: 36px; }
button { width:150px;height:50px; font-size: 24px; }
footer { text-align: right; }
.underTheEarthKai { background-image: radial-gradient(50% 150%, #CCCCCC 5%, #777777 100%); }
background-image: linear-gradient(-173deg, rgba(255,255,255,0.20) 0%, #000000 100%),
linear-gradient(72deg, rgba(255,255,255,0.25) 25%, rgba(0,0,0,0.25) 100%),
radial-gradient(47% 102%, rgba(255,255,255,0.50) 0%, rgba(21,24,32,0.60) 120%);background-blend-mode: multiply; }
#msg { vertical-align:middle; }
.floatleft { float:left; }
.floatright { float:right; }
--></style>
<title>M5Stack TimerCamera</title><link rel="shortcut icon" href="https://hobby-it.com/favicon.ico"></head>
<body class="underTheEarthKai"><center><div id="contents"><header><h1>Web Camera [M5Stack TimerCamera]</h1></header>
<div class="floatright"><button id="capbtn" type="button" onclick="window.open('/reset')">REBOOT</button></div>
<form id="canform" method="get" action="javascript:void(0);">
<div id="contentImg"><img id="live"></div>
<div><div class="floatleft">
<button id="startbtn" onclick="wsConnect()">START</button>
<button id="endbtn" onclick="wsDisconnect()">STOP</button>
</div><div class="floatright">
<button id="clearbtn" onclick="clearimg()">CLEAR</button>
<button id="capbtn" onclick="capture()">ONE-SHOT</button>
</div></div><br><br><br>
<div><font color="red" size=+3><span id="msg">Please press the button</span></font></div>
<footer>@Hobby-IT</footer></form></div></center>

<script language="javascript" type="text/javascript">
function dispMessage(message){ document.getElementById('msg').innerHTML = message;}
function wsConnect(){
 document.getElementById("startbtn").disabled = true;
 var baseHost = document.location.origin;
 var streamUrl = baseHost + ':81/stream';
 document.getElementById('live').src = streamUrl;
 dispMessage('Delivery started');
}
function wsDisconnect(){
 window.stop();
 document.getElementById("startbtn").disabled = false;
 dispMessage('Delivery finished');
}
function clearimg(){
 document.getElementById('live').src='';
 document.getElementById('live').remove();
 var imgTag = document.createElement('img');
 imgTag.id = "live";
 document.getElementById('contentImg').appendChild(imgTag);
 dispMessage('Cleared');
}
function capture(){
 var baseHost = document.location.origin;
 var captureUrl = baseHost + '/capture';
 document.getElementById('live').src = captureUrl;
 dispMessage('I took a picture');
}
</script></body></html>
```

StyleSheet設定
（文字サイズなどデザインに関する設定）

Favicon設定（なくても良い）



HTMLプログラム

Javascriptプログラム

配信サーバへ接続
（配信要求）

**映像は、imgタグのsrcに配信サーバのURLを指定すると送られてくるJPEG画像を順次imgタグに表示してくれる**

配信サーバへ切断要求
（配信停止）

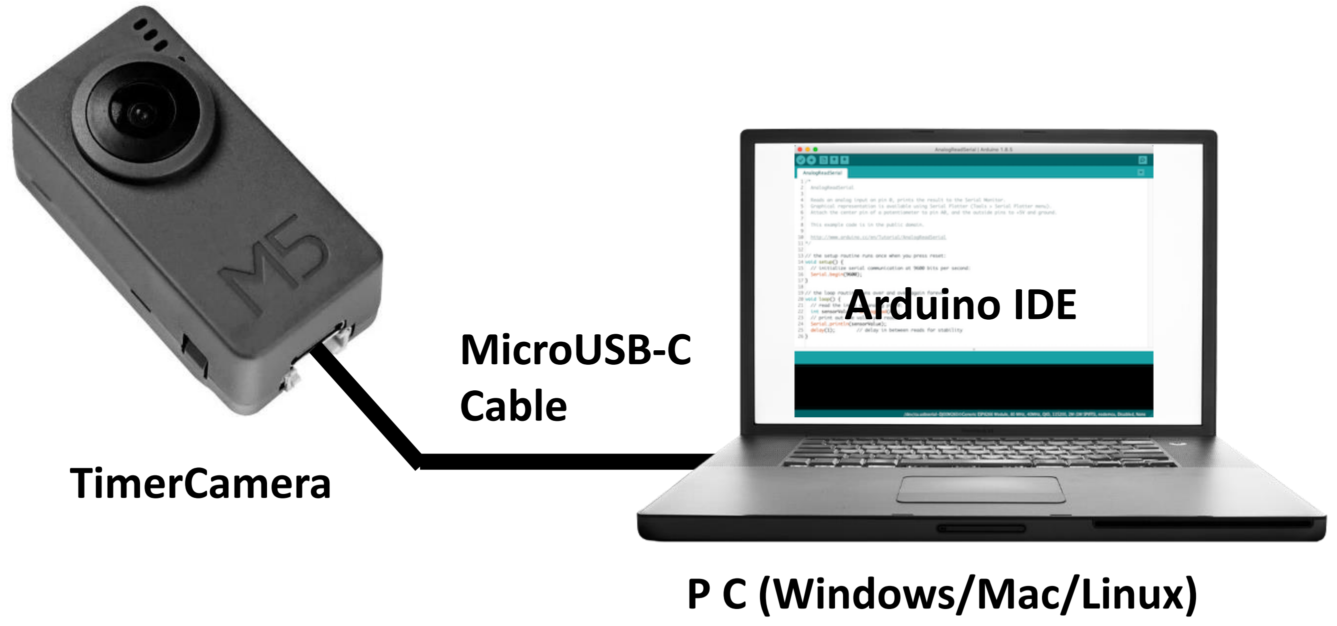
画面クリア

画像取得の要求
（ONE-SHOT処理）

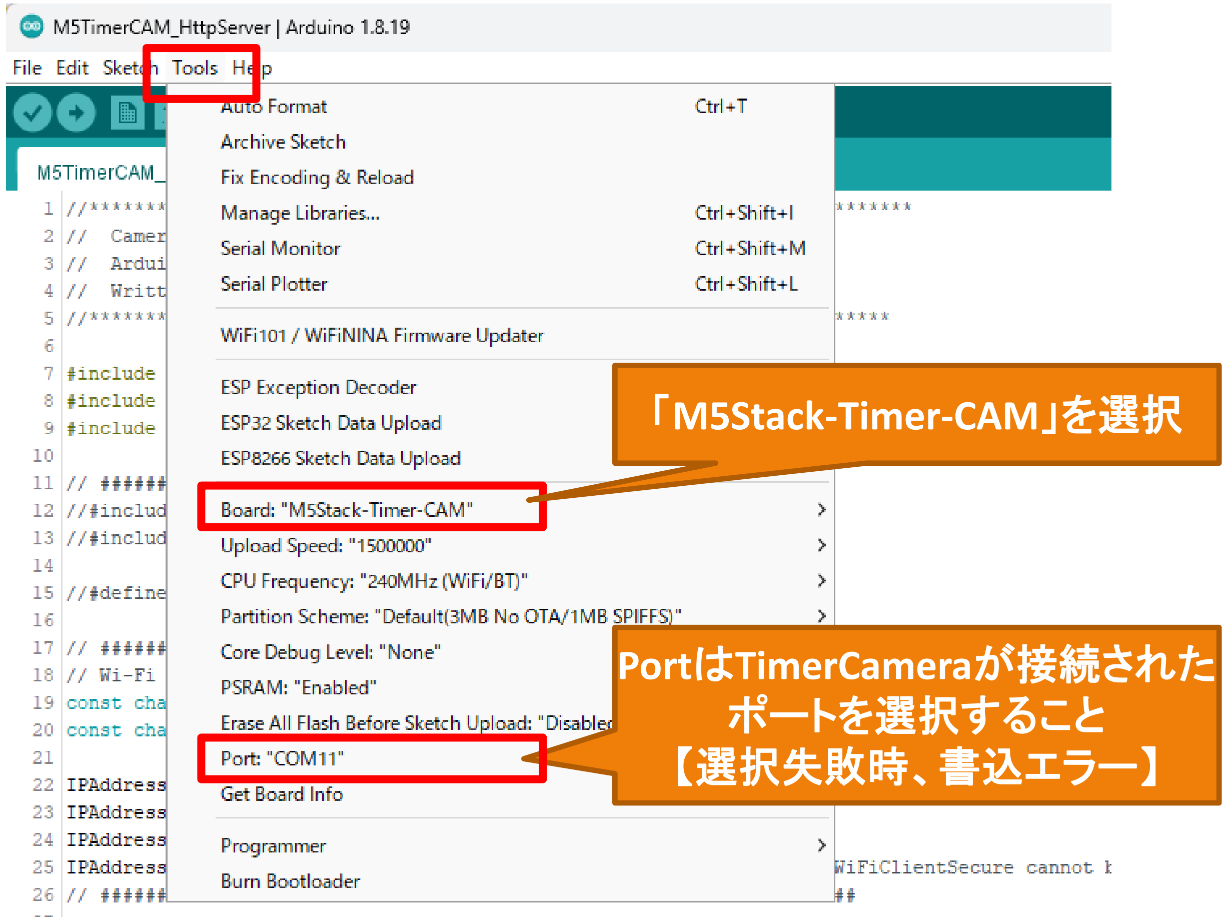
**画像は、imgタグのsrcに配信サーバのURLを指定すると送られてくるJPEG画像をimgタグに表示してくれる**

# ６. プログラム書き込み

1) TimeCameraをマイクロUSB-Cケーブルで接続



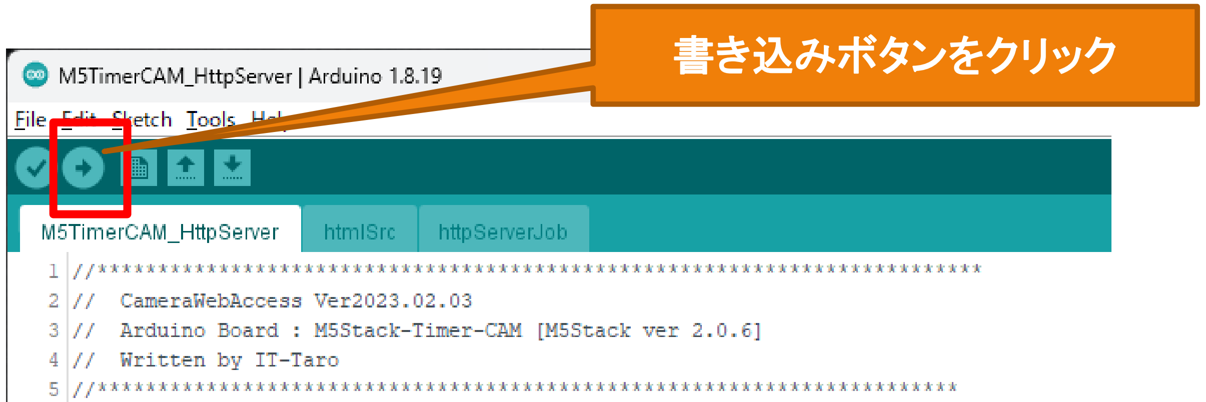
2) ArduinoIDEでプログラムを開き、再度、設定確認。
（プログラムでWi-Fi設定[SSID、IPアドレスなど]は変更しておく。）


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
「M5Stack-Timer-CAM」を選択
PortはTimerCameraが接続されたポートを選択すること【選択失敗時、書込エラー】

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
// ######
```

3) 書き込みボタンをクリック


書き込みボタンをクリック
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
//**********************************************************************
```