Watch Videos on Smartphone [M5Stack TimerCamera]

Implementation of Web server and distribution server
by TimerCamera (ESP32)

Table of Contents 《Watching videos on your smartphone》

1. Equipment selection

2. Development environment

3. Arduino IDE settings

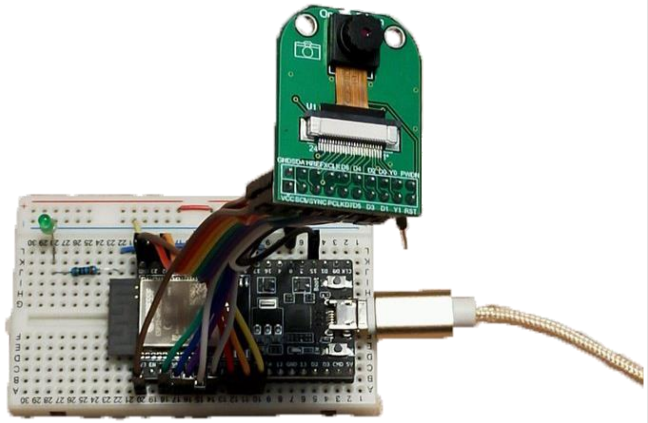
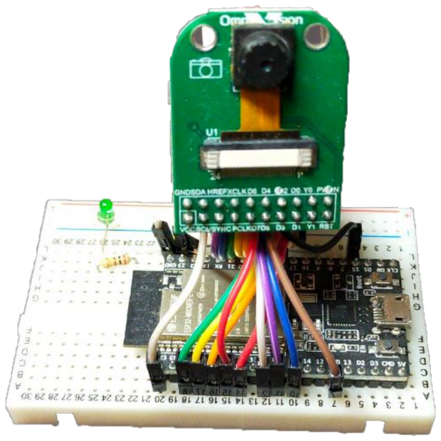
4. Arduino program

5. HTML program

6. program writing

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				TIMER CAMERA F 8M PSRAM ESP32 3M PIXELS OV3660@120° PRODUCT: M5TIMERCAM MODEL: Timer Camera F M5Stack Technology Co.,Ltd.
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 this time

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

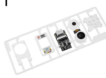
*Please note that the cost varies depending on the season.
* Excel can be downloaded from the Hobby-IT site.

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】

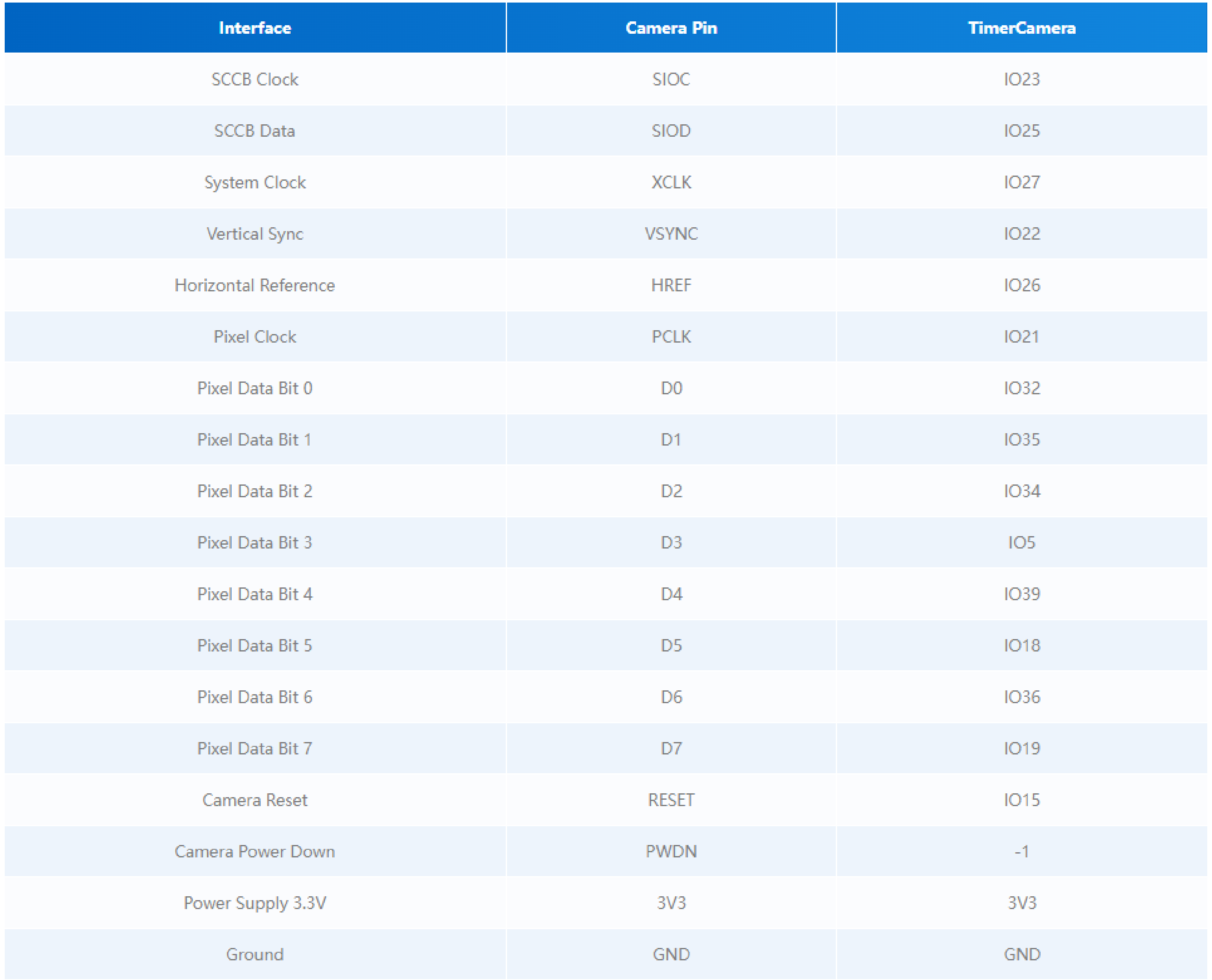
NO	Item	quanti	Image	Item	Shop-URL		Price	Note
1	Timer Camera X	1		ESP32 PSRAM Timer Camera X (OV3660)	https://shop.m5stack.com/collections/m5-cameras	M5Stack	$17.95	viewing angle 66.5°
					https://www.switch-science.com/products/6742	SWITCH SIENCE	2596	
1	Timer Camera F	1		ESP32 PSRAM Timer Camera 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

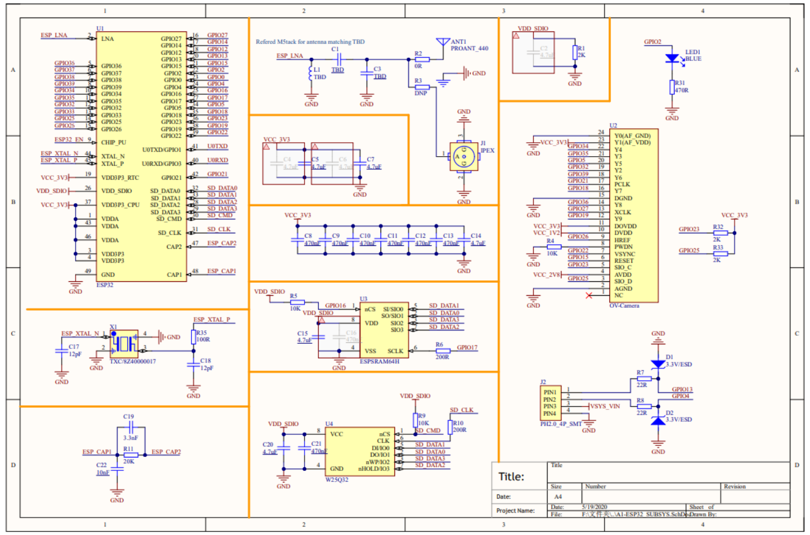
●M5Stack Official TimerCamera Document
https://docs.m5stack.com/en/unit/timercam_x

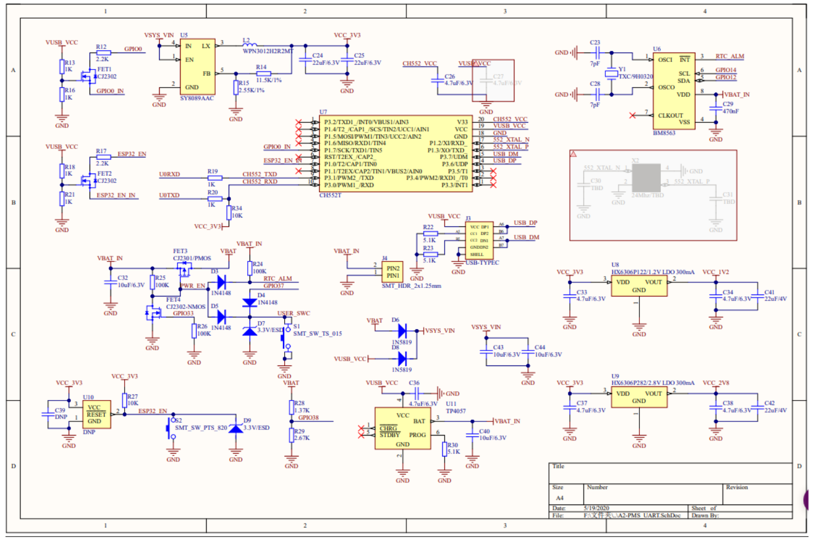
● Pin Map



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

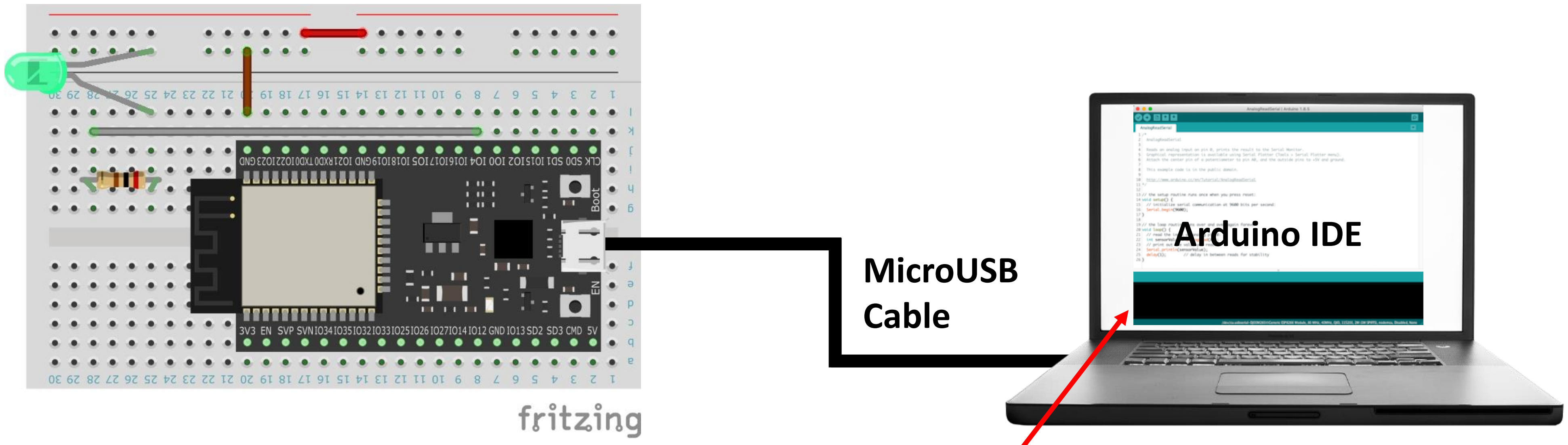
● Schematic





1-2. the development environment “Arduino”

We will use Arduino as the development environment.



MicroUSB Cable

Arduino IDE

ESP32 Development Board

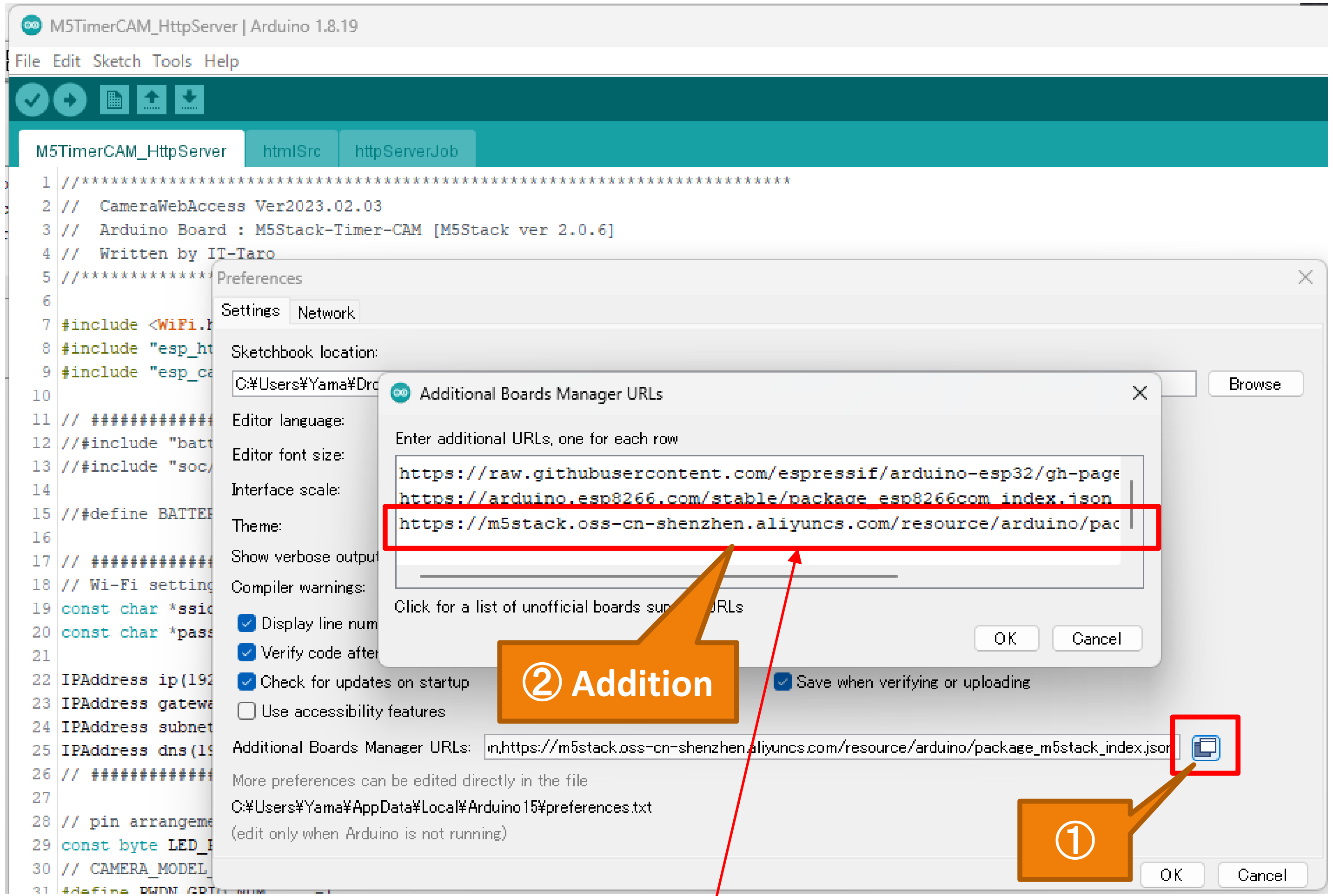
P C (Windows/Mac/Linux)

【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

```
//***************************************************************************
//  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

Sketchbook location: C:¥Users¥Yama¥Dro

Editor language:

Editor font size:

Interface scale:

Theme:

Show verbose output

Compiler warnings:

☑ Display line num

☑ Verify code after

☑ Check for updates on startup

☑ Save when verifying or uploading

☐ Use accessibility features

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 sup URLs

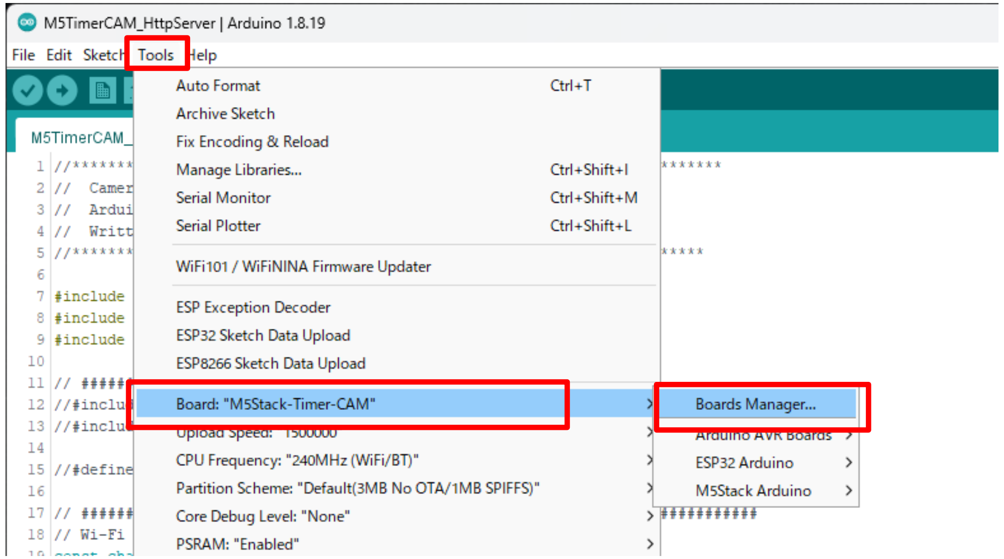
①

② Addition

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
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
Boards Manager...
Arduino AVR Boards
ESP32 Arduino
M5Stack Arduino

Board: "M5Stack-Timer-CAM"

Upload Speed: "1500000"

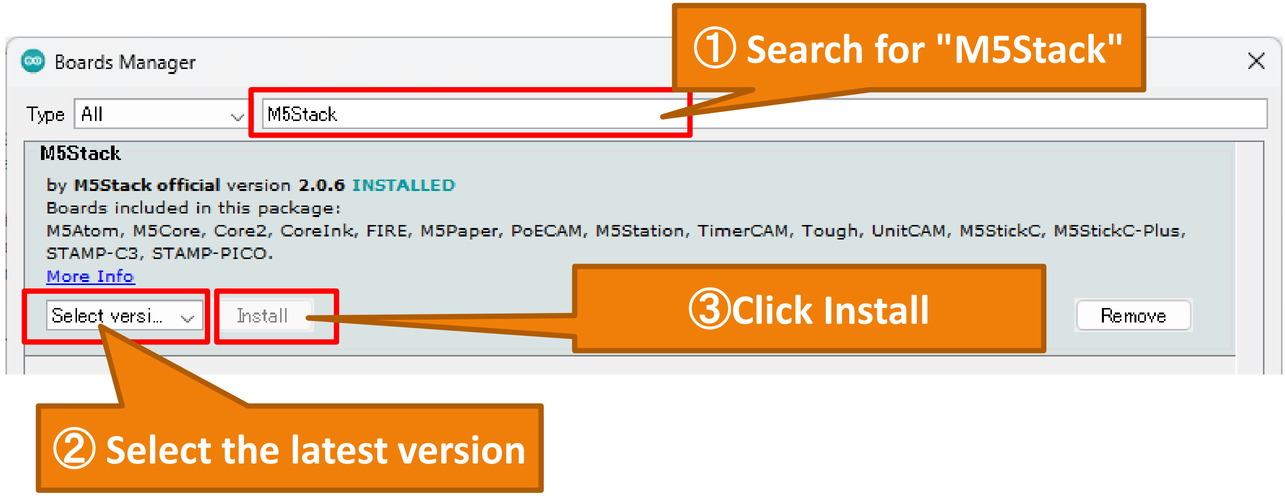
CPU Frequency: "240MHz (WiFi/BT)"

Partition Scheme: "Default(3MB No OTA/1MB SPIFFS)"

Core Debug Level: "None"

PSRAM: "Enabled"

3) Install M5Stack



Boards Manager

Type All　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

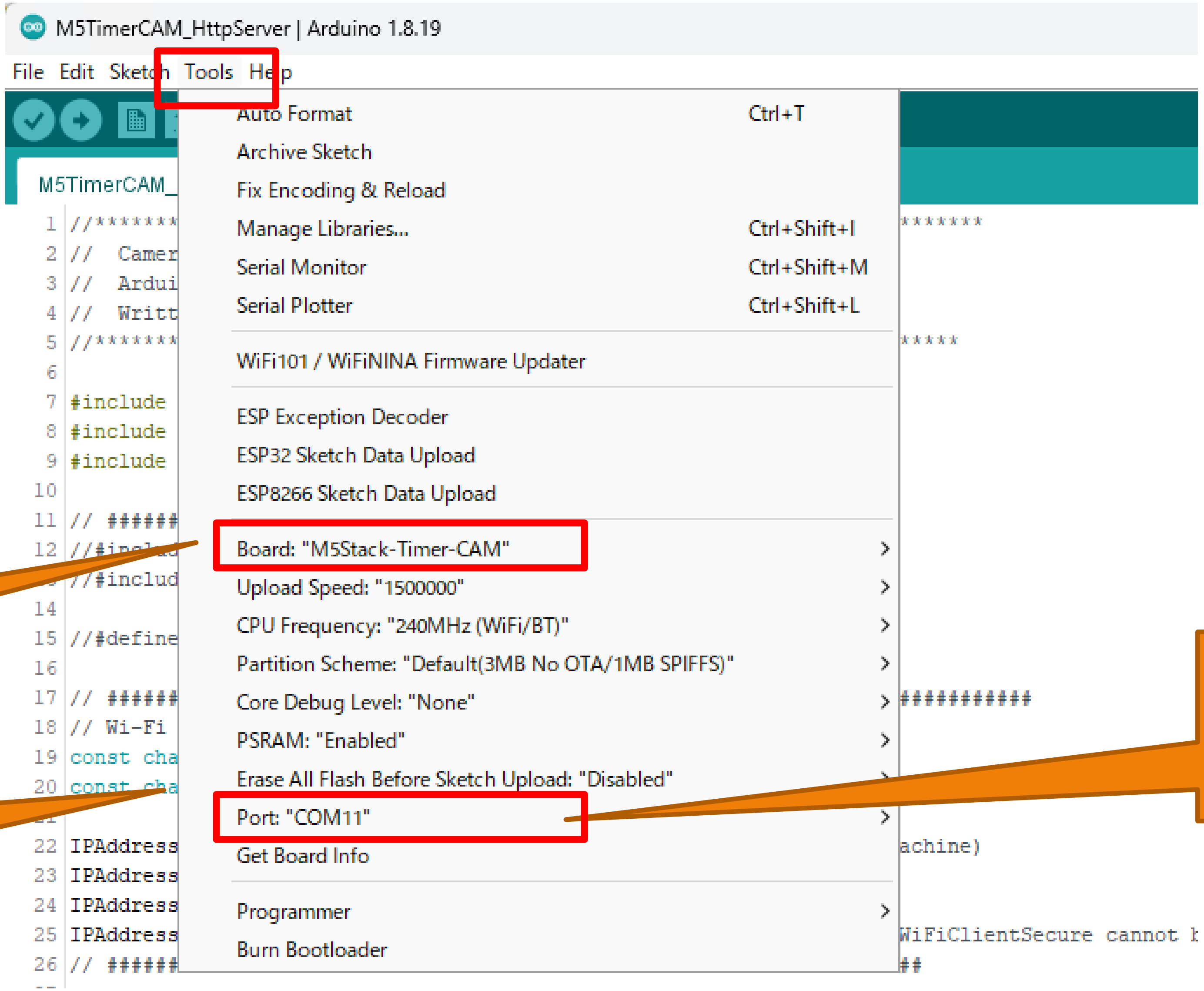
① Search for "M5Stack"

② Select the latest version

③Click Install

3-1. Arduino settings (Board settings)

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


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"

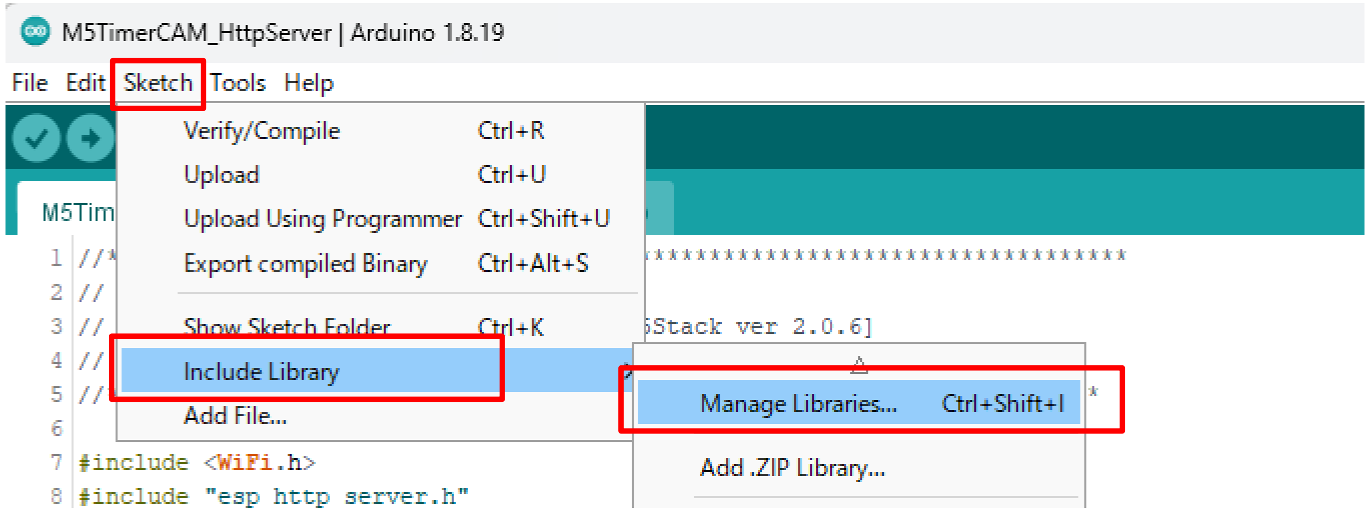
Select "M5Stack-Timer-CAM"

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

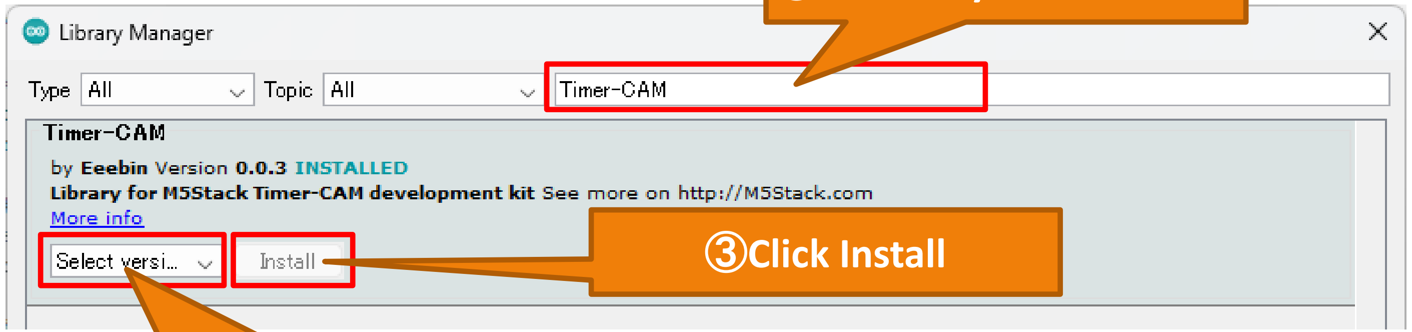


M5TimerCAM_HttpServer | Arduino 1.8.19
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...
M5Tim

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

2) Install "Timer-CAM"

①Search by "Timer-CAM"



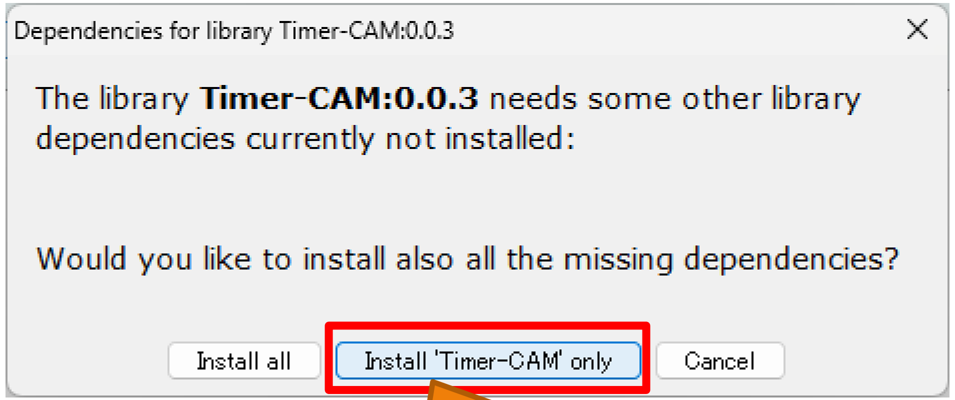
Timer-CAM

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

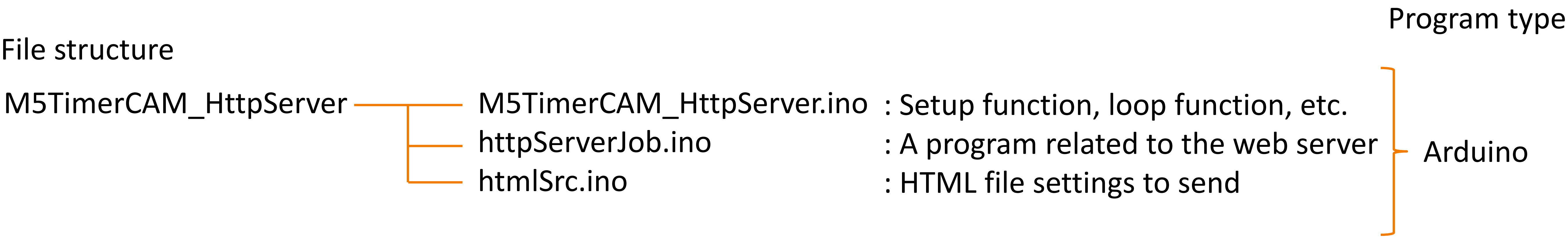
The library **Timer-CAM:0.0.3** needs some other library dependencies currently not installed:

Would you like to install also all the missing dependencies?

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

4-1. Arduino program (file structure)

- File structure



Program type

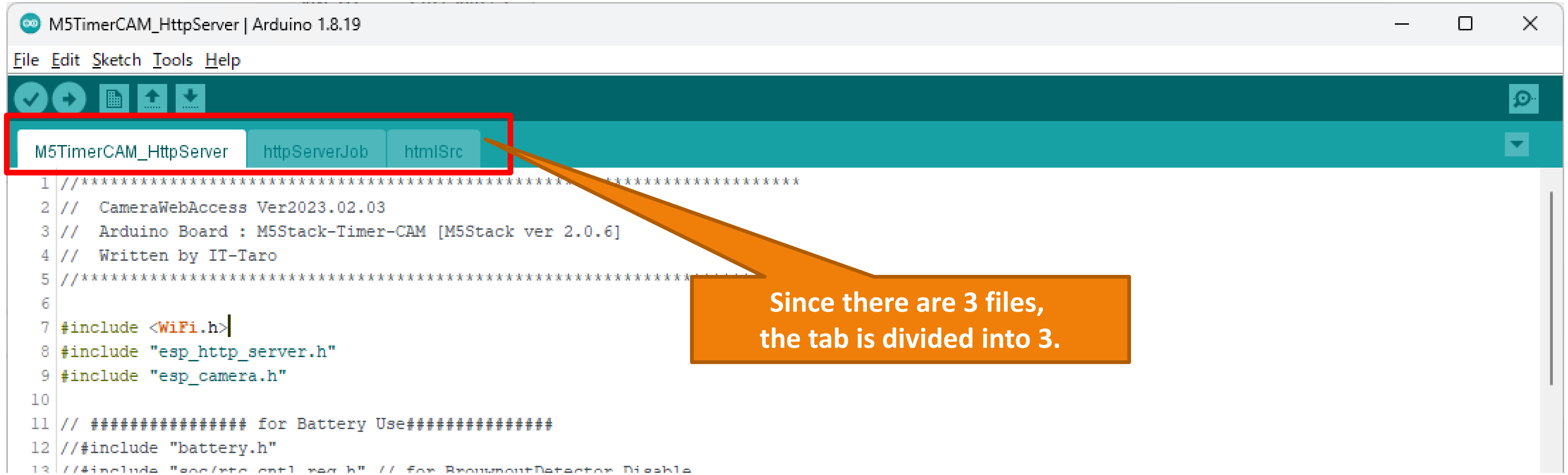
M5TimerCAM_HttpServer

M5TimerCAM_HttpServer.ino : Setup function, loop function, etc.

httpServerJob.ino : A program related to the web server

htmlSrc.ino : HTML file settings to send

Arduino

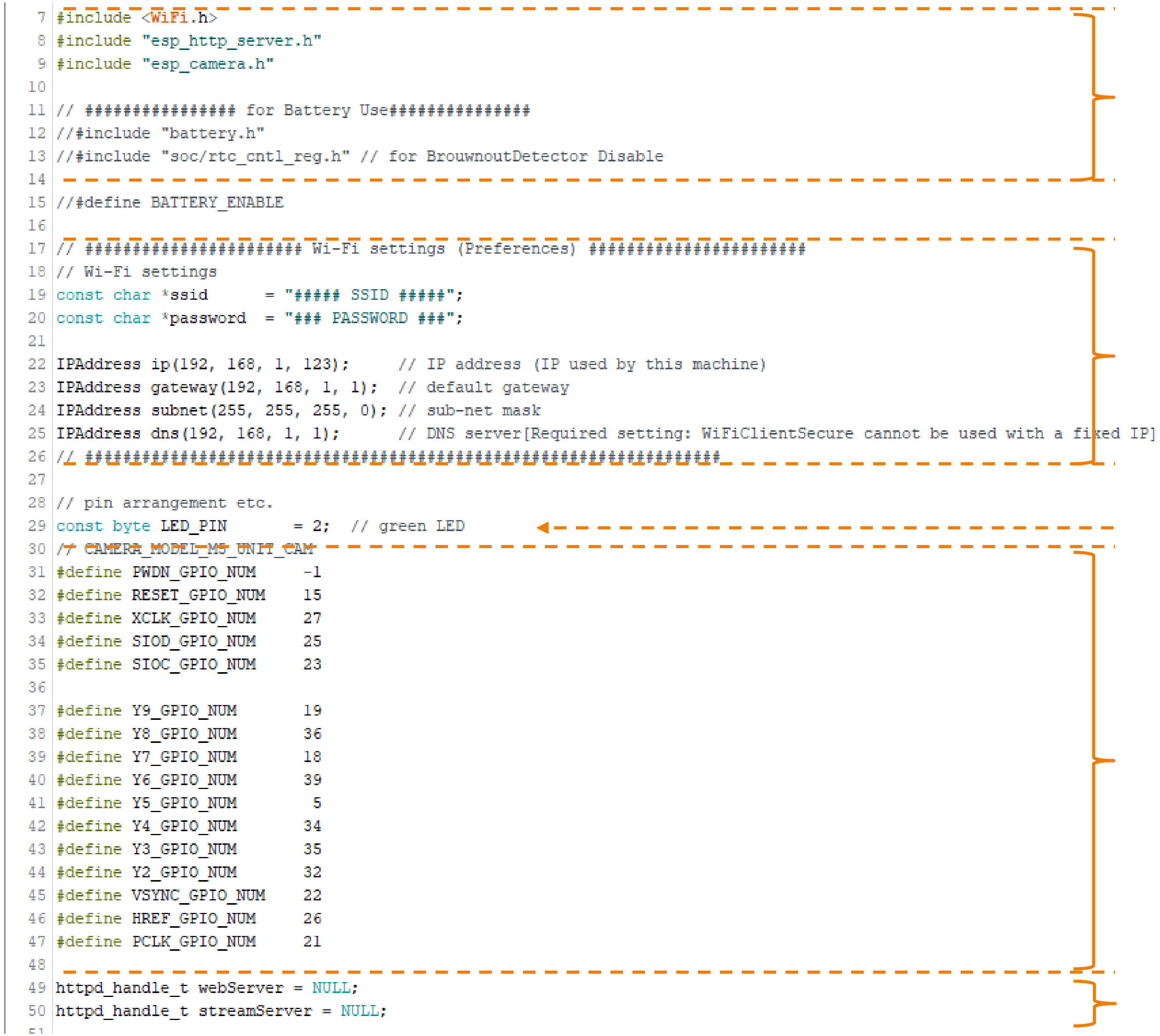


M5TimerCAM_HttpServer | Arduino 1.8.19
File Edit Sketch Tools Help
M5TimerCAM_HttpServer
httpServerJob
htmlSrc

```
//*****************************************************************************************
//  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
```

Since there are 3 files,
the tab is divided into 3.

4-2. Arduino program (global definition)



```
#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 BrounoutDetector 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;
```

loading the library

Wi-Fi settings

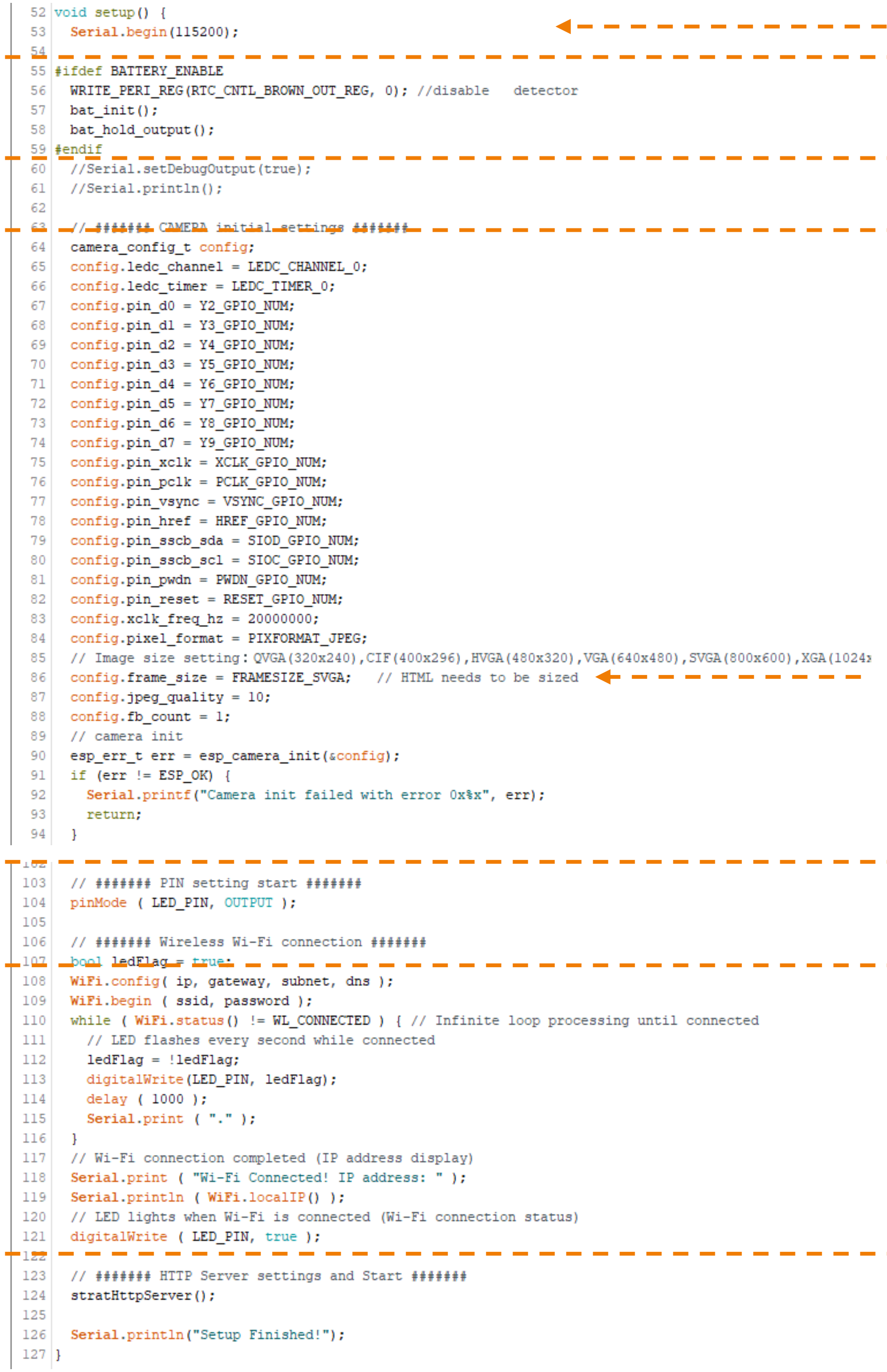
Need to change settings

LED port settings

Camera port settings

Define web server and delivery server

4-3. Arduino program (Setup function)



```
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(1024
  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!");
}
```

Start serial monitor

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

Camera port settings and initialization

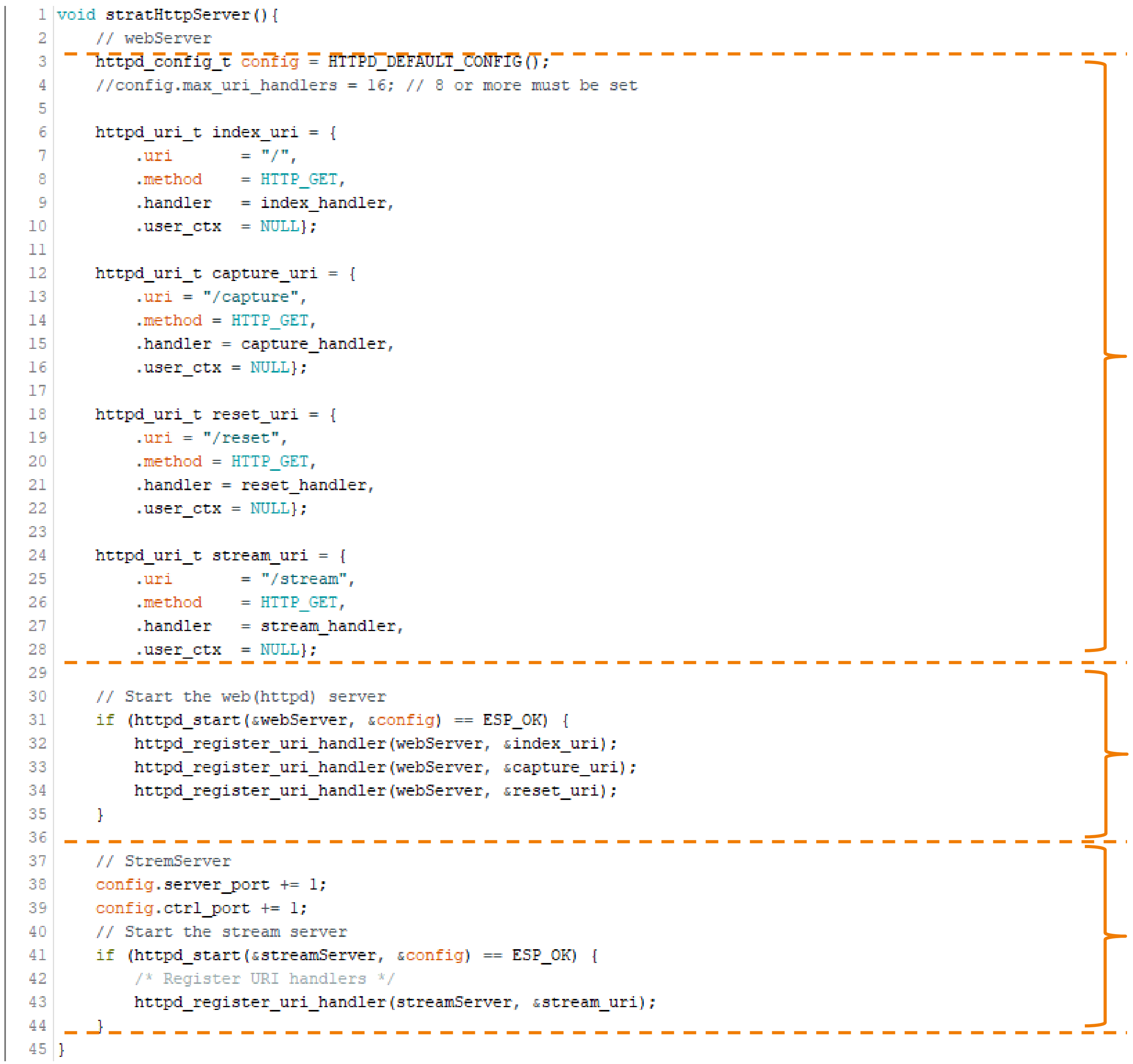
Image size setting

Set the LED port and turn it on

Set Wi-Fi and go to connection

Web server start

4-4. Arduino program (stratHttpServer function)



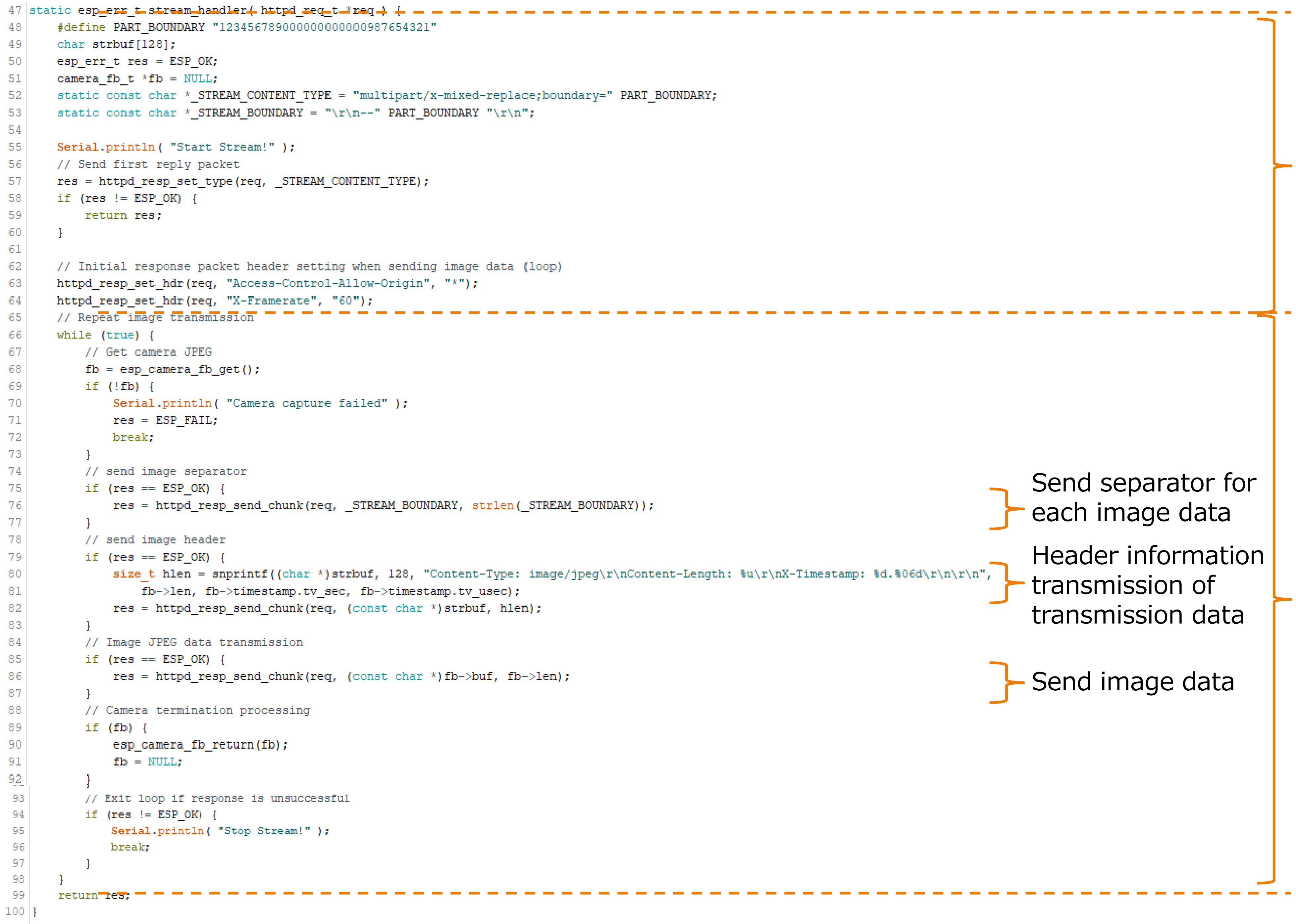
```
void stratHttpServer(){
    // webServer
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    //config.max_uri_handlers = 16; // 8 or more must be set

    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);
    }
}
```

Define access URL and process

Start web server
(Port number: 80)

Start distribution server
(Port number: 81)

4-5. Arduino program (stream_handler function)



```
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;
}
```

Define outgoing header content

Send separator for each image data

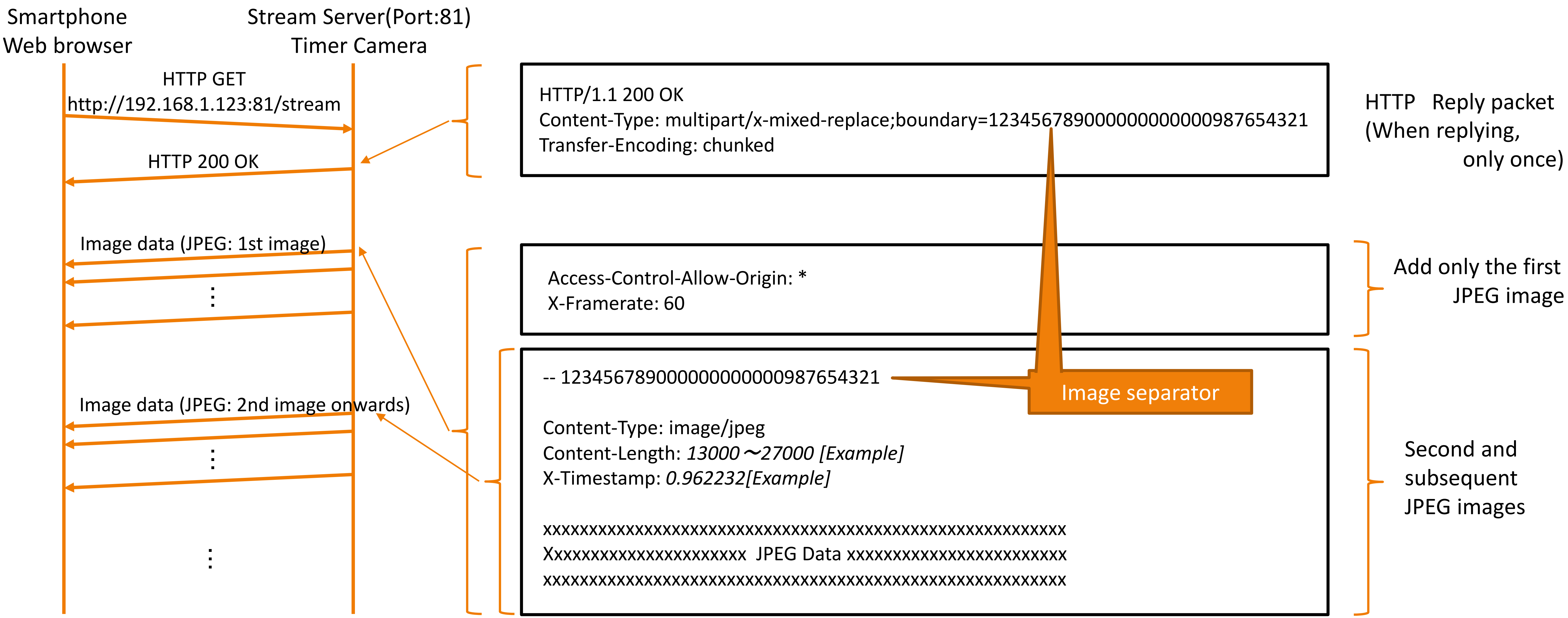
Header information transmission of transmission data

Send image data

Acquire Jpeg image and send to terminal
(Obtain and send one image)

4-5. Arduino program (send data)

- Main communication for video distribution (Motion JPEG)
  Communication such as TCP Ack (response confirmation) is omitted.



Smartphone
Web browser
Stream Server(Port:81)
Timer Camera
HTTP GET
http://192.168.1.123:81/stream
HTTP 200 OK
Image data (JPEG: 1st image)
⋮
Image data (JPEG: 2nd image onwards)
⋮
⋮
Image separator

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

HTTP Reply packet (When replying, only once)

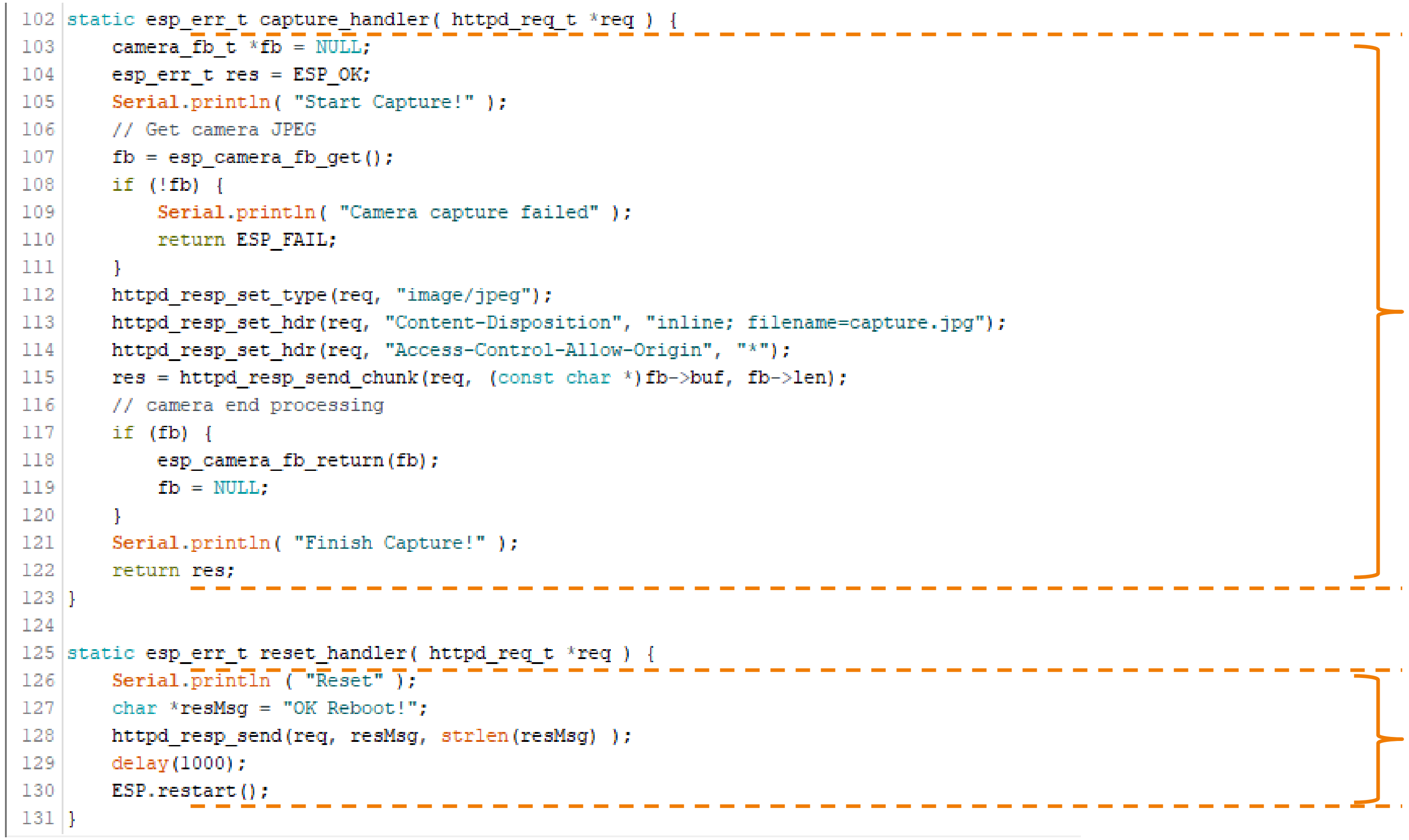
```
Access-Control-Allow-Origin: *
X-Framerate: 60
```

Add only the first JPEG image

```
-- 123456789000000000000987654321

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

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Xxxxxxxxxxxxxxxxxxxxxxx  JPEG Data xxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Second and subsequent JPEG images

4-6. Arduino program (capture_handler, reset_handler functions)



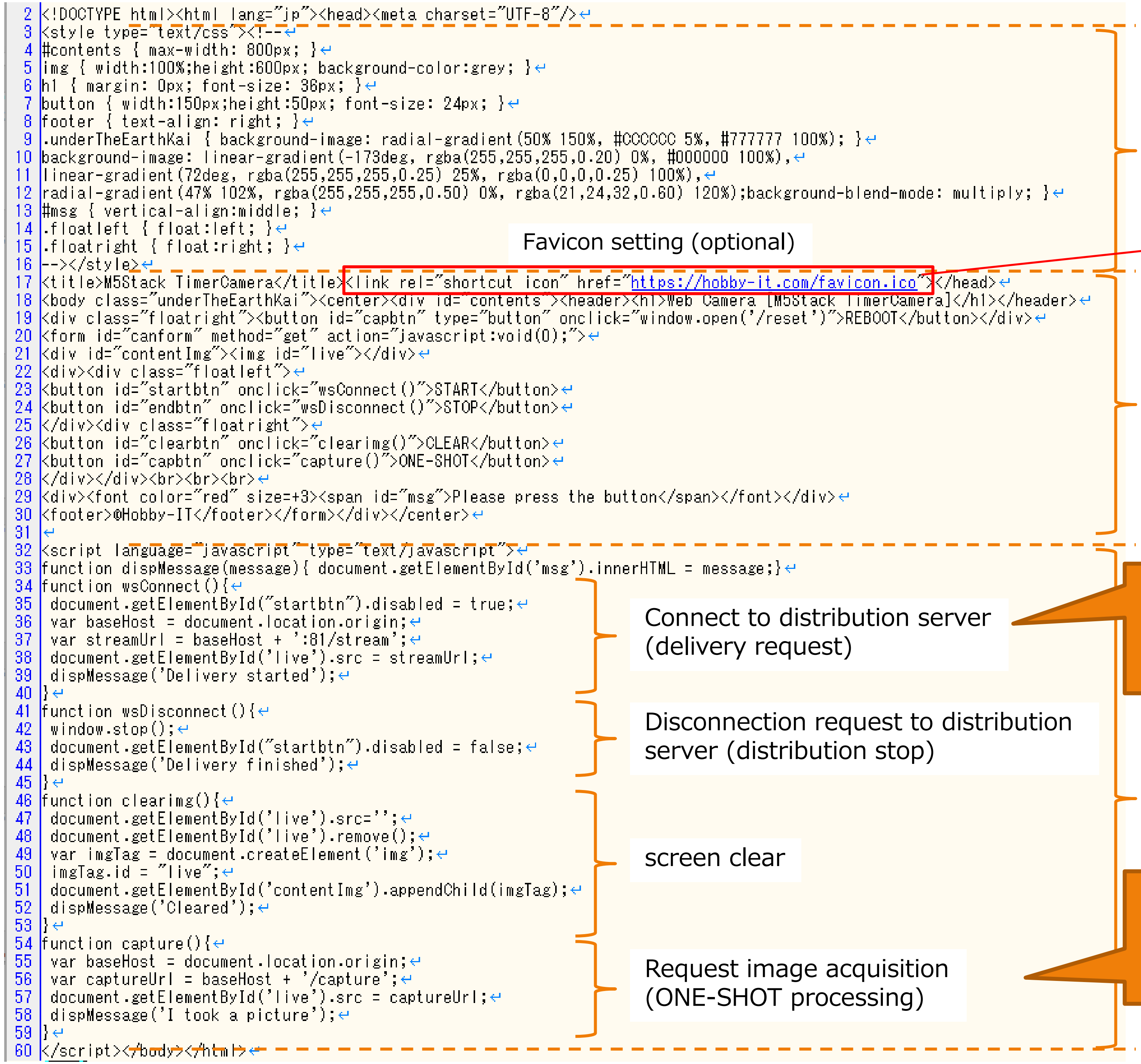
```
static esp_err_t capture_handler( httpd_req_t *req ) {
    camera_fb_t *fb = NULL;
    esp_err_t res = ESP_OK;
    Serial.println( "Start Capture!" );
    // Get camera JPEG
    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println( "Camera capture failed" );
        return ESP_FAIL;
    }
    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();
}
```

Acquire and send images
when ONE-SHOT is executed
(Process only one image)

Restart ESP32 when REBOOT is executed

The "htmlSrc.ino" file only defines variables for HTML, so we will understand it with the HTML program.

5. HTML program



```
<!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>
```

Favicon setting (optional)

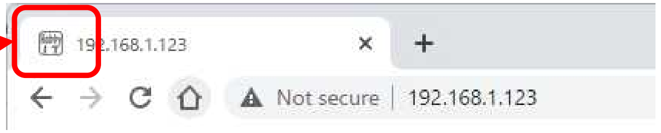
Connect to distribution server (delivery request)

Disconnection request to distribution server (distribution stop)

screen clear

Request image acquisition (ONE-SHOT processing)

StyleSheet settings
(Settings related to design such as font size)



HTML program

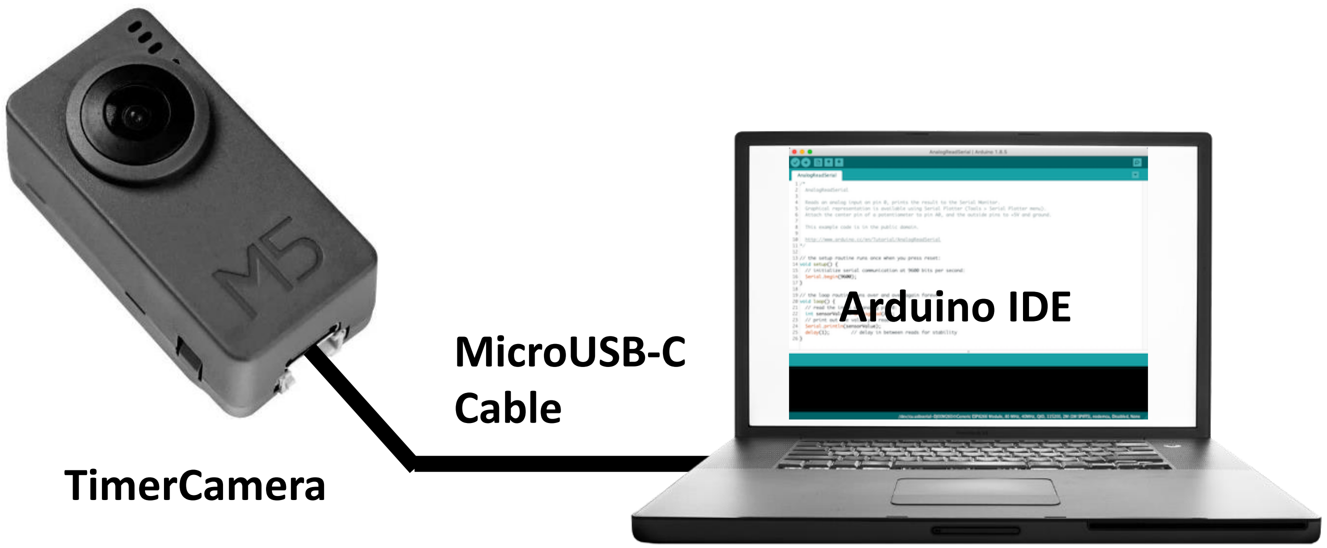
Javascript program

For video, if you specify the URL of the distribution server in the src of the img tag, the JPEG images sent will be displayed sequentially in the img tag.

For images, if you specify the URL of the distribution server in the src of the img tag, the JPEG image sent will be displayed in the img tag.

6. Program writing

1) Connect TimeCamera with micro USB-C cable

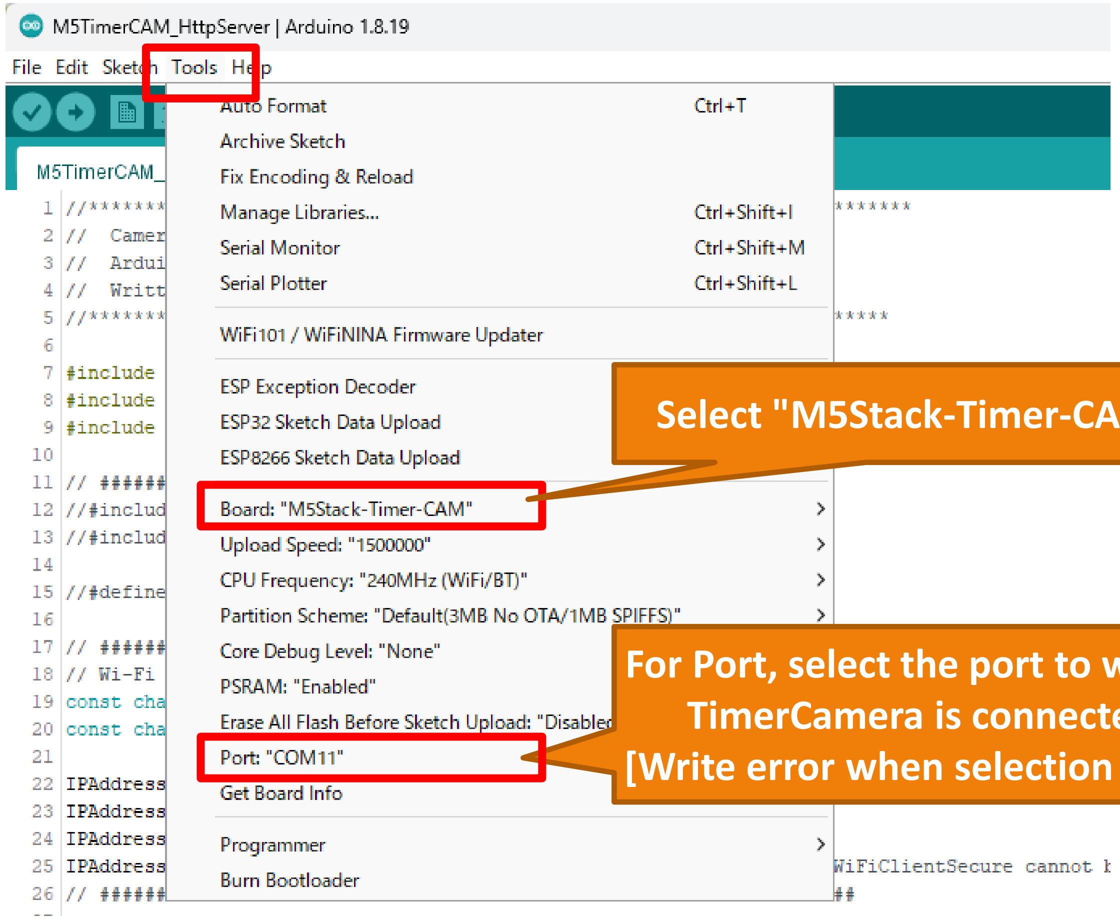

TimerCamera

MicroUSB-C Cable

Arduino IDE

P C (Windows/Mac/Linux)

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

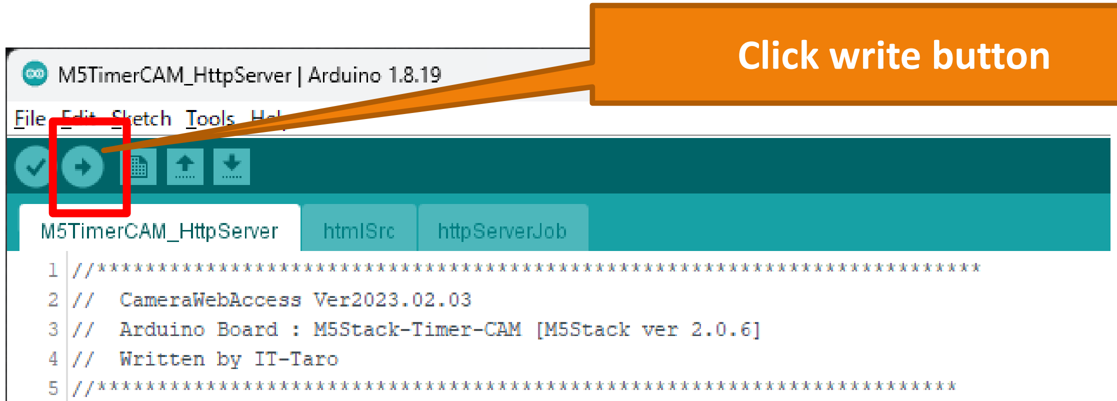

M5TimerCAM_HttpServer | Arduino 1.8.19
File Edit Sketch Tools Help
M5TimerCAM_
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
ESP Exception Decoder
ESP32 Sketch Data Upload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

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