

ESP32 – Cam Code

Esp32-cam | Arduino IDE 2.3.6

File Edit Sketch Tools Help

AI Thinker ESP32-CAM

Esp32-cam.ino

```
1  #include <Arduino.h>
2  #include <ESP32QRCodeReader.h>
3  #include <AsyncTCP.h>
4  #include <ESPAsyncWebServer.h>
5  #include <LittleFS.h>
6  #include "FS.h"
7  #include "SD_MMC.h"
8  #include <time.h>
9  #include <WiFi.h>
10
11 const char* ssid = "YourEnemy";
12 const char* password = "12345678";
13
14 ESP32QRCodeReader reader(CAMERA_MODEL_AI_THINKER);
15
16 long timezone = 0;
17 byte daysavetime = 1;
18
19 const int ledPin = 4;
20 const int triggerPin = 13;
21
22 AsyncWebServer server(80);
23
24 const char* PARAM_INPUT_1 = "qrCode";
25 const char* PARAM_INPUT_2 = "role";
26 const char* PARAM_INPUT_3 = "delete";
27 const char* PARAM_INPUT_4 = "delete-user";
28
29 String inputMessage;
30 String inputParam;
31
32 void onQrCodeTask(void *pvParameters) {
33     struct QRCodeData qrCodeData;
34
35     while (true) {
36         if (digitalRead(triggerPin) == HIGH) {
37             Serial.println("Trigger received from FPGA! Scanning for QR Code...");
38             digitalWrite(ledPin, HIGH);
39
40             if (reader.receiveQrCode(&qrCodeData, 100)) {
41                 //
42             }
43
44             digitalWrite(ledPin, LOW);
45             delay(500);
46         }
47     }
48
49     vTaskDelay(100 / portTICK_PERIOD_MS);
```

Ln 66, Col 2 AI Thinker ESP32-CAM on COM7 [not connected]

Esp32-cam | Arduino IDE 2.3.6

File Edit Sketch Tools Help

AI Thinker ESP32-CAM

Esp32-cam.ino

```
47     vTaskDelay(100 / portTICK_PERIOD_MS);
48 }
49 }
50
51 // Write to the SD card
52 void writeFile(fs::FS &fs, const char * path, const char * message) {
53     Serial.printf("Writing file: %s\n", path);
54
55     File file = fs.open(path, FILE_WRITE);
56     if(!file) {
57         Serial.println("Failed to open file for writing");
58         return;
59     }
60     if(file.print(message)) {
61         Serial.println("File written");
62     } else {
63         Serial.println("Write failed");
64     }
65     file.close();
66 }
67
68 // Append data to the SD card
69 void appendFile(fs::FS &fs, const char * path, const char * message) {
70     Serial.printf("Appending to file: %s\n", path);
71
72     File file = fs.open(path, FILE_APPEND);
73     if(!file) {
74         Serial.println("Failed to open file for appending");
75         return;
76     }
77
78     time_t t = file.getLastWrite();
79     struct tm *tmstruct = localtime(&t);
80
81     char bufferDate[50]; // Adjust buffer size as needed
82     snprintf(bufferDate, sizeof(bufferDate), "%d-%02d-%02d",
83             (tmstruct->tm_year) + 1900,
84             (tmstruct->tm_mon) + 1,
85             tmstruct->tm_mday);
86     char bufferTime[50]; // Adjust buffer size as needed
87     snprintf(bufferTime, sizeof(bufferTime), "%02d:%02d:%02d",
88             tmstruct->tm_hour,
89             tmstruct->tm_min,
90             tmstruct->tm_sec);
91
92     String lastWriteTime = bufferDate;
93     String finalString = String(bufferDate) + "," + String(bufferTime) + "," + String(message) + "\n";
```

Ln 66, Col 2 AI Thinker ESP32-CAM on COM7 [not connected]

Esp32-cam | Arduino IDE 2.3.6

File Edit Sketch Tools Help

AI Thinker ESP32-CAM

Esp32-cam.ino

```
94 Serial.println(lastWriteTime);
95 if(file.print(finalString.c_str())) {
96   Serial.println("Message appended");
97 } else {
98   Serial.println("Append failed");
99 }
100 file.close();
101 }
102
103 // Append data to the SD card
104 void appendUserFile(fs::FS &fs, const char * path, const char * message) {
105   Serial.printf("Appending to file: %s\n", path);
106
107   File file = fs.open(path, FILE_APPEND);
108   if(!file) {
109     Serial.println("Failed to open file for appending");
110     return;
111   }
112
113   String finalString = String(message) + "\n";
114
115   if(file.print(finalString.c_str())) {
116     Serial.println("Message appended");
117   } else {
118     Serial.println("Append failed");
119   }
120   file.close();
121 }
122
123 void deleteFile(fs::FS &fs, const char *path) {
124   Serial.printf("Deleting file: %s\n", path);
125   if (fs.remove(path)) {
126     Serial.println("File deleted");
127   } else {
128     Serial.println("Delete failed");
129   }
130
131   // If the log.txt file doesn't exist, create a file on the SD card and write the header
132   File file = SD_MMC.open("/log.txt");
133   if(!file) {
134     Serial.println("Creating new log.txt file...");
135     writeFile(SD_MMC, "/log.txt", "Date,Time,QR_Code,Role\r\n");
136   }
137   else {
138     Serial.println("log.txt file already exists");
139   }
140 }
```

Ln 66, Col 2 AI Thinker ESP32-CAM on COM7 [not connected]

Esp32-cam | Arduino IDE 2.3.6

File Edit Sketch Tools Help

AI Thinker ESP32-CAM

Esp32-cam.ino

```
140 file.close();
141
142 // If the users.txt file doesn't exist, create a file on the SD card and write the header
143 file = SD_MMC.open("/users.txt");
144 if(!file) {
145   Serial.println("Creating new users.txt file...");
146   writeFile(SD_MMC, "/users.txt", "QR_Code,Role\r\n");
147 }
148 else {
149   Serial.println("users.txt file already exists");
150 }
151 file.close();
152 }
153
154 String processor(const String& var){
155   return String("HTTP GET request sent to your ESP on input field ("
156     + inputParam + ") with value: " + inputMessage +
157     "<br><a href=\"/\"><button class=\"button button-home\">Return to Home Page</button></a>");
158 }
159
160 void deleteLineFromFile(const char* filename, int lineNumber) {
161   File file = SD_MMC.open(filename);
162   if (!file) {
163     Serial.println("Failed to open file for reading.");
164     return;
165   }
166
167   // Read all lines except the one to delete
168   String lines = "";
169   int currentLine = 0;
170   while (file.available()) {
171     String line = file.readStringUntil('\n');
172     if (currentLine != lineNumber) {
173       lines += line + "\n";
174     }
175     currentLine++;
176   }
177   file.close();
178
179   // Write back all lines except the deleted one
180   file = SD_MMC.open(filename, FILE_WRITE);
181   if (!file) {
182     Serial.println("Failed to open file for writing.");
183     return;
184   }
185
186   file.println(lines);
187 }
```

Ln 66, Col 2 AI Thinker ESP32-CAM on COM7 [not connected]

```
Esp32-cam | Arduino IDE 2.3.6
File Edit Sketch Tools Help
AI Thinker ESP32-CAM

Esp32-cam.ino
186   file.print(lines);
187   file.close();
188   Serial.println("Line deleted successfully.");
189 }
190
191 String getRoleFromFile(const char* filename, String qrCode) {
192   File file = SD_MMC.open(filename);
193   if (!file) {
194     Serial.println("Failed to open file for reading.");
195     return "";
196   }
197
198   // Skip the header line
199   file.readStringUntil('\n');
200
201   // Read each line and check for QR Code
202   while (file.available()) {
203     String line = file.readStringUntil('\n');
204
205     int commaIndex = line.indexOf(',');
206     if (commaIndex > 0) {
207       String fileQrCode = line.substring(0, commaIndex);
208       String role = line.substring(commaIndex + 1);
209
210       // Compare qrCode
211       if (fileQrCode == qrCode) {
212         file.close();
213         role.trim(); // Remove any extra spaces or newline characters
214         return role;
215       }
216     }
217   }
218   file.close();
219   return "";
220
221 void initLittleFS() {
222   if(!LittleFS.begin()){
223     Serial.println("An Error has occurred while mounting LittleFS");
224     return;
225   }
226 }
227
228 void initWifi() {
229   // Connect to Wi-Fi
230   Wifi.begin(ssid, password);
231   int connectAttempt = 0;
232   Serial.println("Connecting to Wifi..");
233 }
```

Ln 66, Col 2 AI Thinker ESP32-CAM on COM7 [not connected]

```
Esp32-cam | Arduino IDE 2.3.6
File Edit Sketch Tools Help
AI Thinker ESP32-CAM

Esp32-cam.ino
233 while (Wifi.status() != WL_CONNECTED) {
234   delay(500);
235   Serial.print(".");
236   connectAttempt++;
237   if (connectAttempt == 10){
238     ESP.restart();
239   }
240 }
241 // Print ESP32 Local IP Address
242 Serial.print("\nESP IP Address: ");
243 Serial.println(Wifi.localIP());
244 }
245
246 void initTime() {
247   Serial.println("Initializing Time");
248   struct tm tmstruct;
249   tmstruct.tm_year = 0;
250   getLocalTime(&tmstruct);
251   Serial.printf(
252     "Time and Date right now is : %d-%02d-%02d %02d:%02d:%02d\n", (tmstruct.tm_year) + 1900, (tmstruct.tm_mon) + 1, tmstruct.tm_mday, tmstruct.tm_hou
253     tmstruct.tm_sec
254   );
255 }
256
257 void initSDCard() {
258   if (!SD_MMC.begin("/sdcard", true)) {
259     Serial.println("Card Mount Failed");
260     return;
261   }
262   uint8_t cardType = SD_MMC.cardType();
263
264   if (cardType == CARD_NONE) {
265     Serial.println("No SD card attached");
266     return;
267   }
268
269   Serial.print("SD Card Type: ");
270   if (cardType == CARD_MMC) {
271     Serial.println("MMC");
272   } else if (cardType == CARD_SD) {
273     Serial.println("SDSC");
274   } else if (cardType == CARD_SDHC) {
275     Serial.println("SDHC");
276   } else {
277     Serial.println("UNKNOWN");
278   }
279 }
```

Ln 278, Col 4 AI Thinker ESP32-CAM on COM7 [not connected]

```
Esp32-cam | Arduino IDE 2.3.6
File Edit Sketch Tools Help
AI Thinker ESP32-CAM

Esp32-cam.ino
280 uint64_t cardSize = SD_MMC.cardSize() / (1024 * 1024);
281 Serial.printf("SD Card Size: %lluMB\n", cardSize);
282
283 // If the log.txt file doesn't exist, create a file on the SD card and write the header
284 File file = SD_MMC.open("/log.txt");
285 if(!file) {
286     Serial.println("log.txt file doesn't exist");
287     Serial.println("Creating file...");
288     writeFile(SD_MMC, "/log.txt", "Date,Time,QR_Code,Role\r\n");
289 }
290 else {
291     Serial.println("log.txt file already exists");
292 }
293 file.close();
294
295 // If the users.txt file doesn't exist, create a file on the SD card and write the header
296 file = SD_MMC.open("/users.txt");
297 if(!file) {
298     Serial.println("users.txt file doesn't exist");
299     Serial.println("Creating file...");
300     writeFile(SD_MMC, "/users.txt", "QR_Code,Role\r\n");
301 }
302 else {
303     Serial.println("users.txt file already exists");
304 }
305 file.close();
306 }
307
308 void setup() {
309     Serial.begin(115200); // Initialize serial communication
310     while (!Serial); // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4).
311
312     reader.setup();
313     Serial.println("\nSetup QRCode Reader");
314     reader.beginOnCore(1);
315     Serial.println("Begin on Core 1");
316     xTaskCreate(onQrCodeTask, "onQrCode", 4 * 1024, NULL, 4, NULL);
317
318     initWifi();
319     initLittleFS();
320     configTime(3600 * timezone, daysavetime * 3600, "time.nist.gov", "0.pool.ntp.org", "1.pool.ntp.org");
321     initTime();
322     initSDCard();
323
324     pinMode(ledPin, OUTPUT);
325     digitalWrite(ledPin, LOW);
326     pinMode(triggerPin, INPUT);
```

```
Esp32-cam | Arduino IDE 2.3.6
File Edit Sketch Tools Help
AI Thinker ESP32-CAM

Esp32-cam.ino
326 pinMode(triggerPin, INPUT);
327
328 // Route for root / web page
329 server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
330     request->send(LittleFS, "/full-log.html");
331 });
332 // Route for root /add-user web page
333 server.on("/add-user", HTTP_GET, [](AsyncWebServerRequest *request){
334     request->send(LittleFS, "/add-user.html");
335 });
336 // Route for root /manage-users web page
337 server.on("/manage-users", HTTP_GET, [](AsyncWebServerRequest *request){
338     request->send(LittleFS, "/manage-users.html");
339 });
340
341 // Serve Static files
342 server.serveStatic("/", LittleFS, "/");
343
344 // Loads the log.txt file
345 server.on("/view-log", HTTP_GET, [](AsyncWebServerRequest *request){
346     request->send(SD_MMC, "/log.txt", "text/plain", false);
347 });
348 // Loads the users.txt file
349 server.on("/view-users", HTTP_GET, [](AsyncWebServerRequest *request){
350     request->send(SD_MMC, "/users.txt", "text/plain", false);
351 });
352
353 // Receive HTTP GET requests on <ESP_IP>/get?input=<inputMessage>
354 server.on("/get", HTTP_GET, [](AsyncWebServerRequest *request) {
355     // GET input1 and input2 value on <ESP_IP>/get?input1=<inputMessage1>&input2=<inputMessage2>
356     if (request->hasParam(PARAM_INPUT_1) && request->hasParam(PARAM_INPUT_2)) {
357         inputMessage = request->getParam(PARAM_INPUT_1)->value();
358         inputParam = String(PARAM_INPUT_1);
359         inputMessage += " " + request->getParam(PARAM_INPUT_2)->value();
360         inputParam += " " + String(PARAM_INPUT_2);
361
362         String finalMessageInput = String(request->getParam(PARAM_INPUT_1)->value()) + " " + String(request->getParam(PARAM_INPUT_2)->value());
363         appendUserFile(SD_MMC, "/users.txt", finalMessageInput.c_str());
364     }
365     else if (request->hasParam(PARAM_INPUT_3)) {
366         inputMessage = request->getParam(PARAM_INPUT_3)->value();
367         inputParam = String(PARAM_INPUT_3);
368         if(request->getParam(PARAM_INPUT_3)->value()=="users") {
369             deleteFile(SD_MMC, "/users.txt");
370         }
371         else if(request->getParam(PARAM_INPUT_3)->value()=="log") {
372             deleteFile(SD_MMC, "/log.txt");
```

```
373     }
374 }
375 else if (request->hasParam(PARAM_INPUT_4)) {
376     inputMessage = request->getParam(PARAM_INPUT_4)->value();
377     inputParam = String(PARAM_INPUT_4);
378     deleteLineFromFile("/users.txt", inputMessage.toInt());
379 }
380 else {
381     inputMessage = "No message sent";
382     inputParam = "none";
383 }
384 request->send(LittleFS, "/get.html", "text/html", false, processor);
385 });
386 // Start server
387 server.begin();
388 }
389
390 void loop() {
391
392 }
```

Ln 325, Col 27 AI Thinker ESP32-CAM on COM7 [not connected]