**EXPLAIN AND BUILD A SECURITY SYSTEM THAT DETECTS MOTION, CAPTURES IMAGES, AND SENDS ALERTS TO A MOBILE APP**

Here's a complete **explanation and build guide** for a **security system** that **detects motion, captures images, and sends alerts to a mobile app** using **Arduino, PIR sensor, ESP32-CAM, and Blynk IoT platform**.

**✅ Objective**

Build a **security system** that:

1. **Detects motion** using a PIR sensor.
2. **Captures images** using ESP32-CAM.
3. **Sends an alert** to a mobile app (Blynk) with the captured image.

## **Components Required**

| **Component** | **Quantity** |
| --- | --- |
| ESP32-CAM | 1 |
| PIR Motion Sensor (HC-SR501) | 1 |
| Jumper Wires | As needed |
| Breadboard | 1 |
| 5V Power Supply | 1 |
| Smartphone with Blynk app | 1 |

## 🔌 **Circuit Diagram**

PIR VCC → 5V (ESP32-CAM)

PIR GND → GND (ESP32-CAM)

PIR OUT → GPIO13 (ESP32-CAM)

## 📲 **Mobile App Setup using Blynk (New Blynk Platform)**

1. Download **Blynk IoT** app (iOS or Android).
2. Create a new project.
3. Add:
   * **Notification widget**
   * **Image gallery** or **Web URL widget**
4. Get **Auth Token** from the Blynk dashboard.

## 💻 **Arduino Code (ESP32-CAM)**

## #include "esp\_camera.h"

## #include <WiFi.h>

## #include <BlynkSimpleEsp32.h>

## char auth[] = "Your\_Blynk\_Auth\_Token";

## char ssid[] = "Your\_WiFi\_Name";

## char pass[] = "Your\_WiFi\_Password";

## #define PIR\_PIN 13

## BlynkTimer timer;

## bool motionDetected = false;

## // ESP32-CAM camera config

## void startCamera() {

## camera\_config\_t config;

## config.ledc\_channel = LEDC\_CHANNEL\_0;

## config.ledc\_timer = LEDC\_TIMER\_0;

## config.pin\_d0 = 5;

## config.pin\_d1 = 18;

## config.pin\_d2 = 19;

## config.pin\_d3 = 21;

## config.pin\_d4 = 36;

## config.pin\_d5 = 39;

## config.pin\_d6 = 34;

## config.pin\_d7 = 35;

## config.pin\_xclk = 0;

## config.pin\_pclk = 22;

## config.pin\_vsync = 25;

## config.pin\_href = 23;

## config.pin\_sscb\_sda = 26;

## config.pin\_sscb\_scl = 27;

## config.pin\_pwdn = 32;

## config.pin\_reset = -1;

## config.pin\_xclk = 0;

## config.xclk\_freq\_hz = 20000000;

## config.pixel\_format = PIXFORMAT\_JPEG;

## 

## config.frame\_size = FRAMESIZE\_VGA;

## config.jpeg\_quality = 10;

## config.fb\_count = 1;

## 

## // Initialize camera

## esp\_err\_t err = esp\_camera\_init(&config);

## if (err != ESP\_OK) {

## Serial.printf("Camera init failed with error 0x%x", err);

## return;

## }

## }

## void sendCapturedImage() {

## camera\_fb\_t \*fb = esp\_camera\_fb\_get();

## if (!fb) {

## Serial.println("Camera capture failed");

## return;

## }

## // Send notification

## Blynk.logEvent("motion\_detected", "Motion detected! Sending image...");

## 

## // Upload image to image hosting (like Imgur or web server)

## // For this demo, assume image sent externally and we send a placeholder URL

## Blynk.virtualWrite(V1, "https://your-server.com/captured\_image.jpg");

## esp\_camera\_fb\_return(fb);

## }

## void detectMotion() {

## if (digitalRead(PIR\_PIN) == HIGH) {

## if (!motionDetected) {

## motionDetected = true;

## sendCapturedImage();

## }

## } else {

## motionDetected = false;

## }

## }

## void setup() {

## Serial.begin(115200);

## pinMode(PIR\_PIN, INPUT);

## WiFi.begin(ssid, pass);

## Blynk.begin(auth, ssid, pass);

## startCamera();

## timer.setInterval(1000L, detectMotion);

## }

## void loop() {

## Blynk.run();

## timer.run();

## }

**📤 How the System Works**

1. PIR sensor detects motion.
2. ESP32-CAM captures an image.
3. Image gets uploaded to a server or hosted link.
4. Mobile user receives:
   * **Notification alert**
   * **Link or preview of the image**

**📸 Enhancement Ideas**

* Use **Google Firebase** or **Imgur API** to upload and store images.
*  Add **face recognition** using OpenCV (advanced).
*  Integrate with **Telegram bot** or **WhatsApp alerts**

