# **Muhammad Sultan Ul Arifeen**

📁 **Multi-Controller AI Robot Project: Full System Code & Instructions**

## ✅ SECTION 1: NodeMCU Complete Code

#define BLYNK\_TEMPLATE\_ID "TMPLxxxx"  
#define BLYNK\_TEMPLATE\_NAME "YourTemplate"  
#define BLYNK\_AUTH\_TOKEN "YourAuthToken"  
  
#include **<ESP8266WiFi.h>**  
#include **<BlynkSimpleEsp8266.h>**  
#include **<WiFiManager.h>**  
#include **<Servo.h>**  
  
*// Servo*  
Servo panServo;  
Servo tiltServo;  
  
*// Ultrasonic Sensor*  
const int trigPin = D6;  
const int echoPin = D5;  
long duration;  
int distance;  
  
*// Motors*  
#define motor1Pin1 D1  
#define motor1Pin2 D2  
#define motor2Pin1 D3  
#define motor2Pin2 D4  
  
*// Serial with Arduino*  
#define BAUD\_RATE 9600  
  
*// Blynk buttons*  
BLYNK\_WRITE(V1) { digitalWrite(motor1Pin1, param.asInt()); } *// Forward*  
BLYNK\_WRITE(V2) { digitalWrite(motor1Pin2, param.asInt()); } *// Backward*  
BLYNK\_WRITE(V3) { digitalWrite(motor2Pin1, param.asInt()); } *// Left*  
BLYNK\_WRITE(V4) { digitalWrite(motor2Pin2, param.asInt()); } *// Right*  
  
BLYNK\_WRITE(V10) { *// AI Mode Toggle*  
 int val = param.asInt();  
 Serial.println(val == 1 ? "AI\_MODE\_ON" : "AI\_MODE\_OFF");  
}  
  
BLYNK\_WRITE(V11) { *// Take Photo Trigger*  
 WiFiClient client;  
 HTTPClient http;  
 http.begin(client, "http://192.168.4.120/capture");  
 int httpCode = http.GET();  
 http.end();  
}  
  
BLYNK\_WRITE(V12) { *// Auto Docking Mode*  
 int val = param.asInt();  
 **if** (val == 1) {  
 *// Trigger charging relay or docking logic*  
 }  
}  
  
*// WiFi + Blynk*  
void setupWiFi() {  
 WiFiManager wifiManager;  
 wifiManager.autoConnect("Robot\_AP");  
}  
  
void setup() {  
 Serial.begin(BAUD\_RATE);  
 setupWiFi();  
 Blynk.begin(BLYNK\_AUTH\_TOKEN, WiFi.SSID().c\_str(), WiFi.psk().c\_str());  
  
 panServo.attach(D7);  
 tiltServo.attach(D8);  
  
 pinMode(trigPin, OUTPUT);  
 pinMode(echoPin, INPUT);  
  
 pinMode(motor1Pin1, OUTPUT);  
 pinMode(motor1Pin2, OUTPUT);  
 pinMode(motor2Pin1, OUTPUT);  
 pinMode(motor2Pin2, OUTPUT);  
}  
  
void loop() {  
 Blynk.run();  
  
 digitalWrite(trigPin, LOW);  
 delayMicroseconds(2);  
 digitalWrite(trigPin, HIGH);  
 delayMicroseconds(10);  
 digitalWrite(trigPin, LOW);  
 duration = pulseIn(echoPin, HIGH);  
 distance = duration \* 0.034 / 2;  
 Blynk.virtualWrite(V5, distance);  
  
 **if** (Serial.available()) {  
 String data = Serial.readStringUntil('\n');  
 Blynk.virtualWrite(V6, data); *// TFT Emotion from Arduino*  
 }  
}

## ✅ SECTION 2: ESP32-CAM Complete Code

#include **"esp\_camera.h"**  
#include **<WiFi.h>**  
#include **<ESPAsyncWebServer.h>**  
  
*// Fixed IP config*  
IPAddress local\_IP(192, 168, 4, 120);  
IPAddress gateway(192, 168, 4, 1);  
IPAddress subnet(255, 255, 255, 0);  
  
const char\* ssid = "YourWiFi";  
const char\* password = "YourPassword";  
  
AsyncWebServer server(80);  
  
void startCameraServer();  
  
void setup() {  
 Serial.begin(115200);  
 WiFi.config(local\_IP, gateway, subnet);  
 WiFi.begin(ssid, password);  
 **while** (WiFi.status() != WL\_CONNECTED) delay(1000);  
  
 camera\_config\_t config;  
 *// Camera pin definitions and settings here...*  
 esp\_camera\_init(&config);  
  
 startCameraServer();  
}  
  
void loop() {}  
  
void startCameraServer() {  
 server.on("/capture", HTTP\_GET, [](AsyncWebServerRequest \*request){  
 camera\_fb\_t \*fb = esp\_camera\_fb\_get();  
 request->send\_P(200, "image/jpeg", fb->buf, fb->len);  
 esp\_camera\_fb\_return(fb);  
 });  
 server.begin();  
}

## ✅ SECTION 3: Arduino UNO Code

#include **<SoftwareSerial.h>**  
SoftwareSerial nodeSerial(2, 3); *// RX, TX*  
  
const int obstaclePin = 7;  
const int angryPin = 8; *// Angry gesture trigger*  
  
void setup() {  
 Serial.begin(9600);  
 nodeSerial.begin(9600);  
 pinMode(obstaclePin, INPUT);  
 pinMode(angryPin, INPUT);  
}  
  
void loop() {  
 **if** (digitalRead(obstaclePin) == HIGH) {  
 nodeSerial.println("OBSTACLE");  
 }  
 **if** (digitalRead(angryPin) == HIGH) {  
 nodeSerial.println("CRY");  
 }  
}

## 🧰 Libraries Used

| Library | Platform | Author/Source |
| --- | --- | --- |
| Blynk | NodeMCU | [blynk.io](https://github.com/blynkkk) |
| WiFiManager | NodeMCU | tzapu |
| Servo | NodeMCU/Arduino | Arduino built-in |
| ESPAsyncWebServer | ESP32-CAM | me-no-dev |
| ESP Camera | ESP32-CAM | Espressif |

## 🔌 Pin Connections

| Component | Connected To |
| --- | --- |
| Ultrasonic Trig | NodeMCU D6 |
| Ultrasonic Echo | NodeMCU D5 |
| Servo Pan | NodeMCU D7 |
| Servo Tilt | NodeMCU D8 |
| Motor Driver Pins | NodeMCU D1–D4 |
| Arduino TX | NodeMCU RX (D7) |
| Arduino RX | NodeMCU TX (D6) |
| ESP32-CAM | Independent, WiFi |

## 📲 Blynk Dashboard

| Button | VPin | Purpose |
| --- | --- | --- |
| Forward | V1 | Move Forward |
| Backward | V2 | Move Backward |
| Left | V3 | Turn Left |
| Right | V4 | Turn Right |
| Distance | V5 | Ultrasonic Output |
| Expression | V6 | From Arduino |
| AI Mode | V10 | Toggle AI/Manual |
| Take Photo | V11 | Capture Snapshot |
| Dock Mode | V12 | Trigger Charging |

✅ \*\*Now this doc contains EVERYTHING you need to:

* Upload code
* Connect wires
* Run the bot
* Test advanced features like AI, Face Memory, Emotion, Auto Photo, Blynk alerts, and Auto Docking\*\*