# **Muhammad Sultan Ul Arifeen**

**Project Title: AI-Powered Smart Robot Using ESP32-CAM, NodeMCU, and Arduino UNO**

### 📋 Components Required with Ratings

| Component | Quantity | Voltage/Current Rating | Purpose |
| --- | --- | --- | --- |
| ESP32-CAM | 1 | 3.3V/500mA | Object/face detection, streaming |
| NodeMCU (ESP8266) | 1 | 3.3V/250mA | WiFi control, motor + servo logic |
| Arduino UNO | 1 | 5V/500mA | Optional: extra processing |
| L298N Motor Driver | 1 | 12V/2A per channel | Drives 4 DC motors |
| DC Gear Motors | 4 | 6–12V/300–500mA | Locomotion |
| SG90 Servo Motor | 2 | 5V/250mA | Pan-tilt for camera |
| Ultrasonic Sensor (HC-SR04) | 1 | 5V/15mA | Distance measurement & obstacle avoid |
| IR Sensor | 1 | 3.3V/5V | Proximity detection |
| Power Bank / Battery | 1 | 5V/12V (depends on motors used) | Power supply |

### 🧠 System Overview

#### ESP32-CAM

* Face & object recognition using OpenCV and ESP32-CAM libraries
* Live video stream to local IP (used by Blynk)
* AI pet mode: follows detected face/object using tracking logic

#### NodeMCU (Central Controller)

* Uses Blynk for remote control & data transmission
* Receives commands from ESP32-CAM (over UART or GPIOs)
* Controls 4 DC motors via L298N
* Moves servo motors for pan-tilt
* Handles ultrasonic sensor and IR sensor
* Reports real-time data to Blynk dashboard
* Automatically starts WiFiManager captive portal if no WiFi is stored

#### Arduino UNO

* Optional: Can be used to offload sensor control or backup logic if NodeMCU gets overloaded

### 🔌 Wiring Connections (Basics)

#### ESP32-CAM to Servos:

* GND → Servo GND
* 5V (via regulator) → Servo VCC
* GPIO12 → Servo (Pan)
* GPIO13 → Servo (Tilt)

#### ESP32-CAM UART to NodeMCU (for commands):

* ESP32 GPIO1 (TX) → NodeMCU RX
* ESP32 GPIO3 (RX) ← NodeMCU TX

#### NodeMCU to L298N:

* D1 → IN1
* D2 → IN2
* D5 → IN3
* D6 → IN4
* ENA, ENB → connected to PWM pins via jumpers

#### NodeMCU to Ultrasonic:

* D7 → Trig
* D8 → Echo

#### NodeMCU to IR Sensor:

* A0 → IR OUT

### 💻 Code: NodeMCU (WiFi + Blynk + Motor + Sensor Logic)

#define BLYNK\_TEMPLATE\_ID "TMPLxxxx"  
#define BLYNK\_TEMPLATE\_NAME "Robot Controller"  
#define BLYNK\_AUTH\_TOKEN "YourBlynkToken"  
  
#include **<ESP8266WiFi.h>**  
#include **<BlynkSimpleEsp8266.h>**  
#include **<WiFiManager.h>**  
#include **<Servo.h>**  
  
Servo servoX;  
Servo servoY;  
  
#define trigPin D7  
#define echoPin D8  
  
#define ENA D3  
#define IN1 D1  
#define IN2 D2  
#define IN3 D5  
#define IN4 D6  
#define ENB D4  
  
long duration;  
int distance;  
  
void setup() {  
 Serial.begin(9600);  
  
 WiFiManager wifiManager;  
 wifiManager.autoConnect("NodeMCU\_Robot\_Setup");  
  
 Blynk.begin(BLYNK\_AUTH\_TOKEN, WiFi.SSID().c\_str(), WiFi.psk().c\_str());  
  
 pinMode(trigPin, OUTPUT);  
 pinMode(echoPin, INPUT);  
 pinMode(IN1, OUTPUT); pinMode(IN2, OUTPUT);  
 pinMode(IN3, OUTPUT); pinMode(IN4, OUTPUT);  
 pinMode(ENA, OUTPUT); pinMode(ENB, OUTPUT);  
  
 servoX.attach(D0);  
 servoY.attach(D9);  
}  
  
void loop() {  
 Blynk.run();  
 measureDistance();  
}  
  
void measureDistance() {  
 digitalWrite(trigPin, LOW);  
 delayMicroseconds(2);  
 digitalWrite(trigPin, HIGH);  
 delayMicroseconds(10);  
 digitalWrite(trigPin, LOW);  
 duration = pulseIn(echoPin, HIGH);  
 distance = duration \* 0.034 / 2;  
 Blynk.virtualWrite(V1, distance);  
 **if**(distance < 20){ stopMotors(); }  
}  
  
void stopMotors() {  
 digitalWrite(IN1, LOW); digitalWrite(IN2, LOW);  
 digitalWrite(IN3, LOW); digitalWrite(IN4, LOW);  
}

### 📸 Code: ESP32-CAM Face/Object Detection + UART Commands

(*Use Arduino IDE with ESP32 board support*)

#include **<esp\_camera.h>**  
#include **<WiFi.h>**  
#include **<esp\_http\_server.h>**  
  
#define TXD 1  
#define RXD 3  
  
void setup() {  
 Serial.begin(9600);  
 Serial2.begin(9600, SERIAL\_8N1, RXD, TXD);  
 camera\_config\_t config;  
 *// configure camera...*  
 esp\_err\_t err = esp\_camera\_init(&config);  
 **if** (err != ESP\_OK) **return**;  
}  
  
void loop() {  
 *// Assume face detected logic*  
 bool faceDetected = **true**;  
 **if**(faceDetected){  
 Serial2.println("TRACK\_FACE");  
 }  
 delay(1000);  
}

### 📲 Blynk Dashboard Setup

**1. Create a New Template:**

* Name: “Robot Controller”
* Add Device → ESP8266

**2. Add Widgets:**

| Widget | Virtual Pin | Function |
| --- | --- | --- |
| Gauge | V1 | Distance display |
| Joystick | V2, V3 | Movement control (map to motors) |
| Button | V4 | AI Mode Toggle |
| Video Stream | N/A | Enter ESP32-CAM IP (e.g., [http://192.168.1.X](http://192.168.1.x)) |

**3. Auth Token:** Copy and paste into NodeMCU code.

**4. Save and Deploy:** Blynk IoT dashboard will auto-sync with phone and PC.

### 📡 Final Notes

* Make sure ESP32-CAM is powered with external 5V/2A supply.
* Use logic level shifter if connecting ESP32 TX → NodeMCU RX (because of 3.3V/5V mismatch).
* Secure servos with rubber dampers to avoid overcurrent draw.