# **Muhammad Sultan Ul Arifeen**

# 📘 AI-Powered Multicontroller Robot Project

### 🛠️ Project Overview

An advanced, modular robot with multiple microcontrollers (ESP32-CAM, NodeMCU, Arduino UNO) that supports AI-based interaction, object tracking, face detection, emotion-based visual feedback (via TFT display), object avoidance, Wi-Fi streaming, cloud alerts, Blynk control, face memory, and offline fun-style commands. Designed for DIY makers and researchers.

## 🔩 Hardware Components & Purpose

| Component | Quantity | Role in Project |
| --- | --- | --- |
| ESP32-CAM | 1 | Face/Object detection and video streaming |
| NodeMCU (ESP8266) | 1 | Central Controller: receives data, handles Wi-Fi, motors, Blynk dashboard |
| Arduino UNO | 1 | Peripheral control: auxiliary sensor input, fallback logic |
| L298N Motor Driver | 1 | Controls robot DC motors |
| 4 DC Motors | 4 | Robot movement (left and right wheels) |
| Servo Motors | 2 | Pan-tilt movement for camera mount |
| Ultrasonic Sensor (HC-SR04) | 1 | Object detection & avoidance |
| IR Sensor | 1 | Proximity detection/gesture trigger |
| TFT Touchscreen (2.4” or 3.5”) | 1 | Displays animated eyes & feedback |
| Power Bank (12V or 7.4V battery + regulator) | 1 | Portable power supply |
| OLED (optional) | 1 | Debugging info if needed |

## 🌐 Communication Setup

### NodeMCU ↔ ESP32-CAM: Wi-Fi

* **ESP32-CAM uses fixed IP (e.g., 192.168.4.50)**
* ESP32 streams live video through its own MJPEG server
* NodeMCU fetches ESP32-CAM IP or pushes IP to Blynk for remote viewing

### NodeMCU ↔ Arduino UNO: Serial UART

* Tx-Rx connected via SoftwareSerial
* Arduino sends sensor readings or executes fallback logic

## 📷 ESP32-CAM Streaming Code Snippet

#include **"esp\_camera.h"**  
#include **<WiFi.h>**  
  
const char\* ssid = "YourSSID";  
const char\* password = "YourPassword";  
const IPAddress local\_IP(192, 168, 4, 50);  
const IPAddress gateway(192, 168, 4, 1);  
const IPAddress subnet(255, 255, 255, 0);  
  
void startCameraServer();  
  
void setup() {  
 WiFi.config(local\_IP, gateway, subnet);  
 WiFi.begin(ssid, password);  
 **while** (WiFi.status() != WL\_CONNECTED) {  
 delay(500);  
 }  
 startCameraServer();  
}  
  
void loop() {}

Access stream at: http://192.168.4.50

## 🧠 Smart AI Behavior

### 🎭 Emotion via TFT Display (Blue Animated Eyes)

* **Excited**: If ESP32-CAM recognizes known face
* **Fear**: If object detected too close
* **Neutral/Idle**: No movement, default blinking animation

### 🧏 Sound Detection

* Use sound sensor to trigger responses like “What was that?”

### 😠 Offline Fun Commands

* “Come here” = robot follows face
* “Angry face” = robot makes sad eyes or cries

## 🎮 Blynk Dashboard (Cloud)

* **2 Joysticks**
  + One for robot movement (forward/backward/turn)
  + One for pan/tilt servo of ESP32-CAM
* **Photo Button**: Trigger ESP32-CAM to capture and send photo to Google Drive
* **Dock Button**: Sends robot to docking station for recharge
* **Alert System**: Sends Blynk notifications on events like unknown face detection, obstacle alert, low battery

## ☁️ Auto Wi-Fi Manager (NodeMCU)

#include **<WiFiManager.h>**  
WiFiManager wm;  
  
void setup() {  
 **if** (!wm.autoConnect("Robot\_AP")) {  
 ESP.restart();  
 }  
 *// WiFi credentials saved for future boot*  
}

## 🧠 Face Memory & Auto Click

* ESP32-CAM recognizes known faces and stores them (using internal SPIFFS or SD card)
* On recognition: can auto click photo and trigger animation

## ⚡ Auto Docking (Manual Trigger)

* When Dock button on Blynk is pressed, robot navigates to predefined IR beacon or station
* Once docked, enters charging state (uses relay or MOSFET based charging switch)

## 🛡️ Safety and Redundancy

* If NodeMCU loses Wi-Fi, robot enters offline safe mode
* Arduino handles local object detection and emergency stop

## 🔚 Summary

This project simulates a pet-like AI robot controlled via the cloud (Blynk), local voice/facial gestures, and sensor fusion. It’s modular, extensible, and designed with real-time control and feedback mechanisms.