This is a fantastic **wildlife conservation project** that uses technology to mitigate human-elephant conflicts. Below is an overview of how you can **design and implement** this system.

**Project Overview: Elephant Detection and Deterrent System**

**Objective**

Detect vibrations indicating potential elephant movement, verify presence using a **camera-based AI system**, and trigger deterrent mechanisms like a **buzzer** and **bee sound playback** while notifying authorities via a **GSM module**.

**Hardware Components**

1. **Vibration Sensor** → Detects ground vibrations.
   * Example: **Geophone sensor (SM-24) or MEMS accelerometer (ADXL345)**
2. **ESP32-S3 + ESP32-CAM** → AI-based object detection.
   * **ESP32-CAM** captures images when vibration is detected.
   * **ESP32-S3** runs **YOLOv8** lightweight model for elephant recognition.
3. **Buzzer Module** → Alerts locals when elephants are detected.
4. **Audio Playback Module** → Plays **bee sounds** (Elephants fear bees!).
   * Example: **DFPlayer Mini MP3 module**
5. **GSM Module (SIM800L or SIM7600)** → Sends notifications via SMS or calls.
6. **Power Supply & Backup Battery** → Ensures continuous operation.

**Implementation Steps**

**1. Detect Vibrations**

* The **geophone** or **accelerometer** senses ground vibrations.
* The **ESP32-S3** reads sensor values to determine elephant movement thresholds.
* If a vibration exceeds a predefined threshold → **Trigger camera module.**

**2. Capture and Analyze Image**

* **ESP32-CAM** takes an image when vibration is detected.
* **ESP32-S3 runs YOLOv8** (or TinyML model) to **identify elephants**.
* If an elephant is detected → **Trigger deterrent actions.**

**3. Sound Alarm & Bee Sound Playback**

* **Buzzer** activates to alert people nearby.
* **MP3 Module (DFPlayer Mini)** plays **bee buzzing sounds** to repel elephants.
* Audio plays for a certain period before resetting.

**4. Notify Authorities via GSM**

* If an elephant is detected, **ESP32-S3 sends an alert** via GSM:
  + SMS format:
  + Elephant detected at [Location]. Please take action!
  + Call or alert tones can also be triggered.

**5. System Optimization**

* **Add solar panel power supply** for off-grid use.
* **Deploy multiple nodes** in different locations for a broader monitoring network.
* **Integrate LoRa** for long-range communication instead of GSM in some cases.

## ****Hardware Requirements****

### ****1. Core Processing & AI Unit****

✅ **ESP32-S3-N8R2** – Runs AI-based object detection and system control.  
✅ **ESP32-CAM (OV2640 or OV5640)** – Captures images for YOLO elephant detection.  
✅ **MicroSD Card** – Stores images and logs.

### ****2. Vibration Detection System****

✅ **Geophone Sensor (SM-24) or MEMS Accelerometer (ADXL345)** – Detects ground movement.  
✅ **ADS1115 ADC Module** (if needed) – Converts analog signals for ESP32-S3-N8R2.

### ****3. AI-Based Object Detection****

✅ **TENSORFLOW, YOLOv8-Tiny or MobileNetV2** – Optimized for embedded AI on ESP32-S3-N8R2.

### ****4. Alarm & Deterrent System****

✅ **Buzzer Module (5V or 12V Piezo)** – Sounds alarm.  
✅ **DFPlayer Mini MP3 Module** – Plays **bee buzzing sounds** to deter elephants.  
✅ **Speaker (3W or 5W)** – Amplifies deterrent sounds.

### ****5. Display & User Interface****

✅ **160x80 0.96’ LCD Display (SPI)** – Shows system status, detection alerts, and debug information.

### ****6. Wireless Communication & Alerts****

✅ **SIM800L GSM Module** – Sends SMS alerts.   
✅ **Wi-Fi & MQTT** – Alternative for online data transmission via Blynk.

### ****7. Power Supply & Backup****

✅ **12V Solar Panel + Charge Controller** – Provides off-grid operation.  
✅ **LiFePO4 Battery (12V, 10Ah or higher)** – Ensures continuous power.  
✅ **Buck Converter (5V/3.3V)** – Regulates voltage.

### ****8. Additional Components****

✅ **Relay Module (5V or 12V)** – Controls sirens or external alarms.  
✅ **Weatherproof Enclosure** – Protects components from environmental damage.  
✅ **PCB or Perfboard** – Assembles circuits neatly.

### ****Power Requirements Including ESP32-CAM****

#### **1. 3.3V Buck Converter Power Needs**

| **Component** | **Estimated Current (mA)** |
| --- | --- |
| ESP32-S3-N8R2 | ~250mA (active mode) |
| ADXL345 Accelerometer | ~40mA |
| 160x60 LCD (spi) | ~20mA |
| GSM Module (SIM800L) | ~500mA (during transmission) |
| **Total (Peak)** | **~810mA (~0.8A)** |

#### **1. 4V Buck Converter Power Needs**

| **Component** | **Estimated Current (mA)** |
| --- | --- |
| GSM Module (SIM800L) | ~2A (during transmission) |

**Recommended Buck Converter:**

* **4V, 3A** (to ensure stable operation)

#### **2. 5V Buck Converter Power Needs**

| **Component** | **Estimated Current (mA)** |
| --- | --- |
| ESP32-CAM | ~180mA (idle), ~310mA (flash on) |
| DFPlayer Mini MP3 | ~150mA |
| Speaker (3W) | ~500mA |
| Relay Module | ~100mA |
| **Total (Peak)** | **~1060mA (~1.1A)** |

**Recommended Buck Converter:**

* **5V, 2A** (to handle peak loads)

### ****Key Considerations****

* **ESP32-CAM power consumption varies** depending on whether the flash is on (~310mA) or off (~180mA).
* **Peak GSM transmission** can spike power usage, so a **buffered power supply** is recommended.
* **Ensure stable voltage regulation** to prevent ESP32-CAM resets due to power fluctuations.

Steps to train a **TensorFlow Lite model** for **elephant detection**.

**1. Gather and Label Training Data**

* Collect images of **elephants** and **non-elephant objects**.
* Use **Roboflow** or **LabelImg** to annotate images with bounding boxes.
* Convert data to **TFRecord format** for TensorFlow training.

**2. Train the Object Detection Model**

* Use **TensorFlow Object Detection API** or **TFLite Model Maker**.
* Choose a model like **MobileNet SSD** or **EfficientDet-Lite**.
* Train the model using **Google Colab** or a local machine.
* Youtube model training video https://www.youtube.com/watch?v=yqkISICHH-U

**3. Convert to TensorFlow Lite**

* Optimize the trained model for **embedded devices**.
* Convert it to **TensorFlow Lite (TFLite)** format.
* Apply **quantization** to reduce model size for ESP32-S3.

**4. Deploy on ESP32-S3**

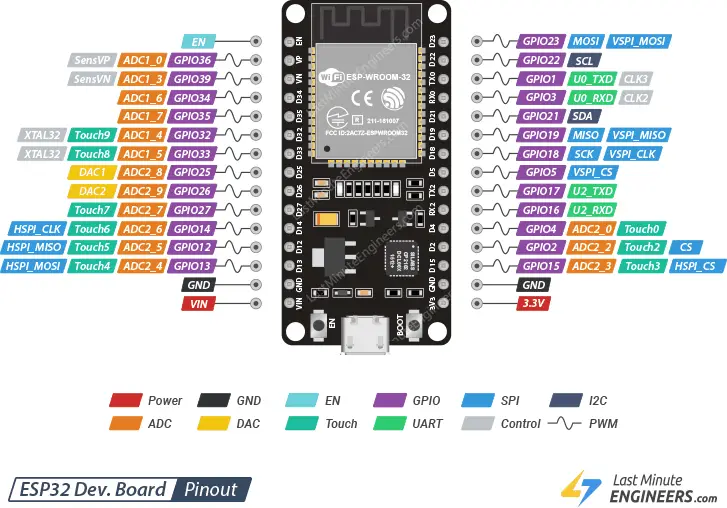
* Load the **TFLite model** onto ESP32-S3.
* Use **TensorFlow Lite Micro** to run inference on captured images.
* Process images locally and classify objects.

**5. Trigger Alerts Based on Detection**

* If an elephant is detected, send an **SMS alert via SIM800L**.
* Update **Blynk** with detection results.
* Activate **buzzer or speaker** to warn nearby areas.

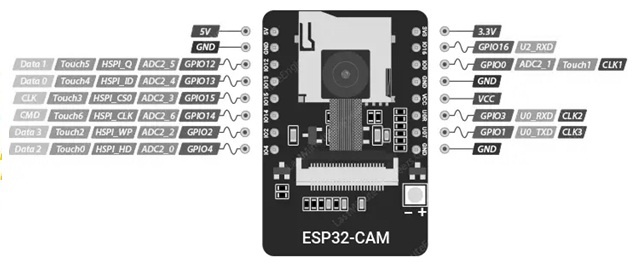
You can follow [this TensorFlow Lite training guide](https://colab.research.google.com/github/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/blob/master/Train_TFLite2_Object_Detction_Model.ipynb) or [this tutorial](https://blog.roboflow.com/how-to-train-a-tensorflow-lite-object-detection-model/) for detailed steps.

ESP32S3 PINOUT

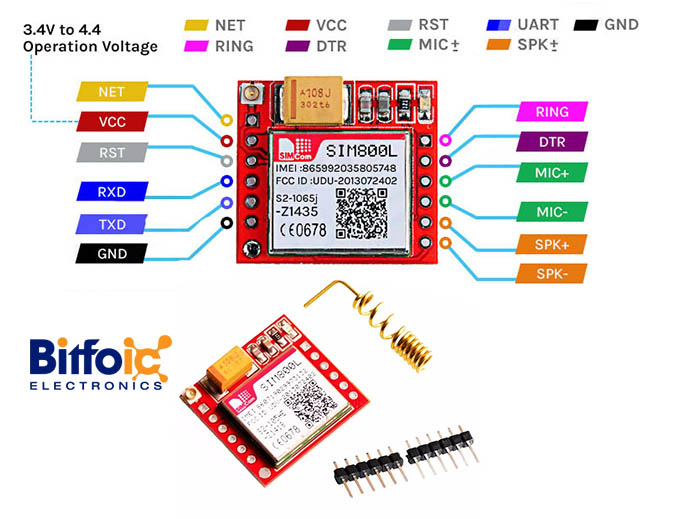


https://lastminuteengineers.com/esp32-pinout-reference/

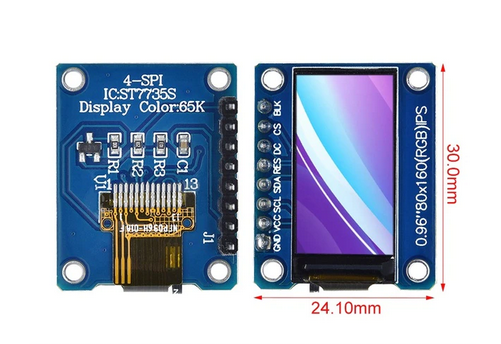
ESP32-CAM PINOUT



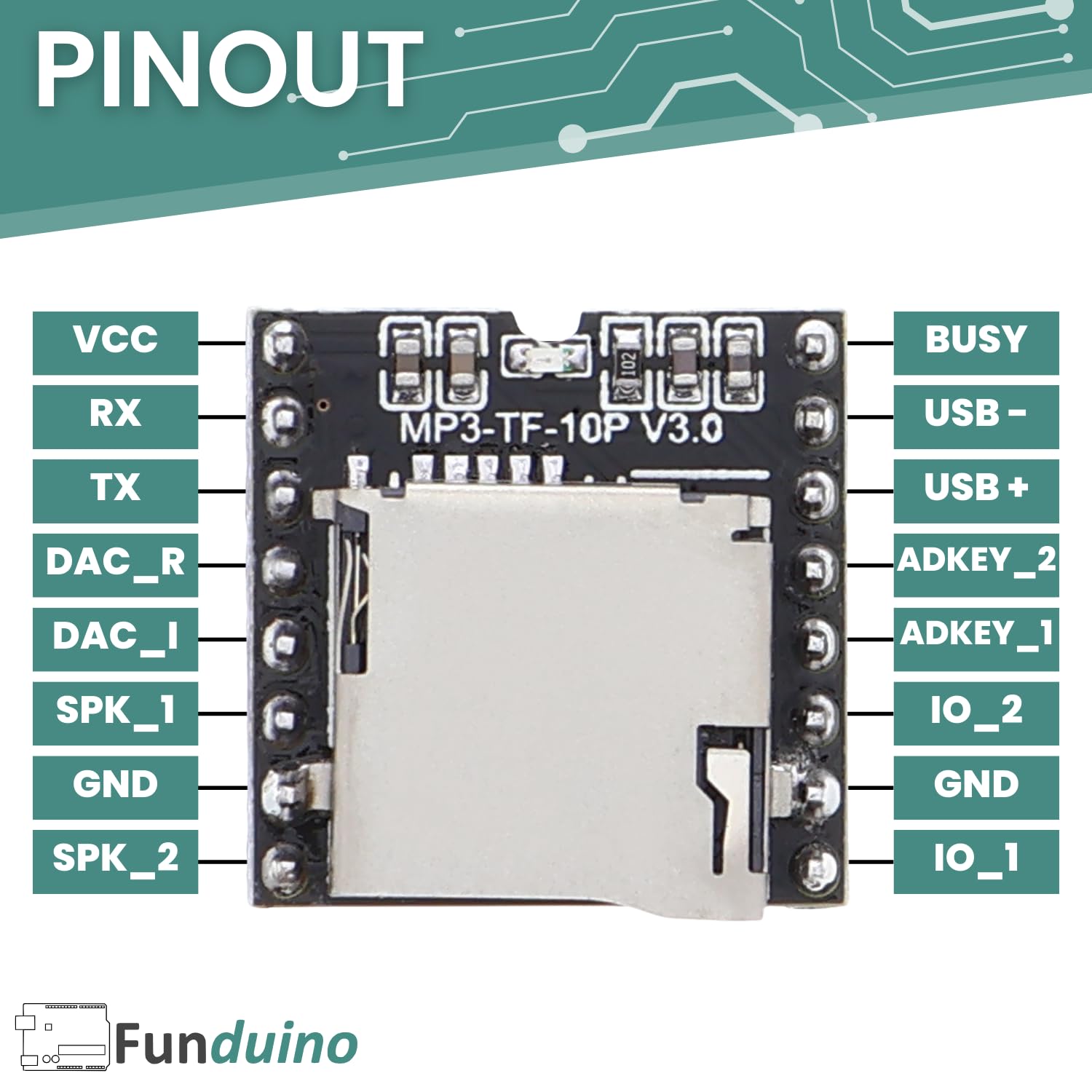
SIM800L PINOUT



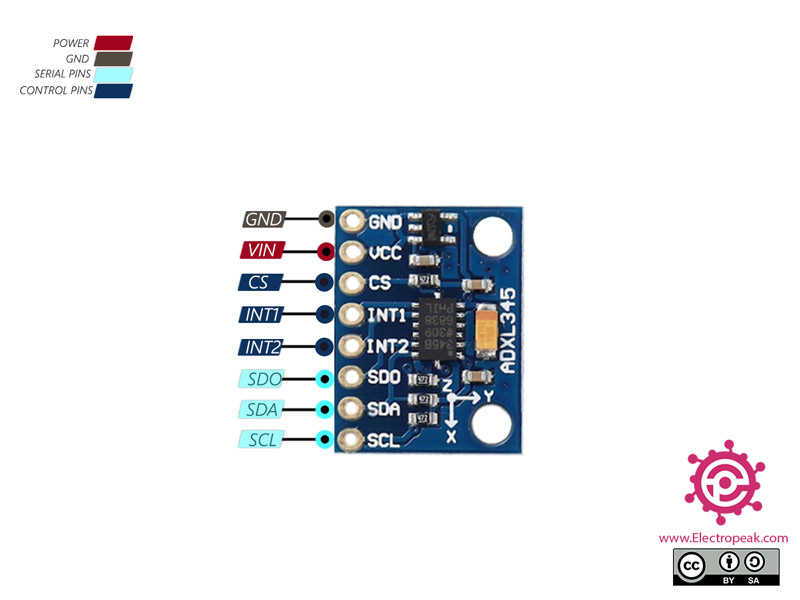
160x80 LCD PINOUT



Mp3-tf-16p v3.0 dfplayer



ADXL345 pinout



**Connection Table**

| **Device** | **Signal/Pin Name** | **ESP32 Pin Number** | **Notes / Direction** |
| --- | --- | --- | --- |
| **LCD (ST7735)** | SCL | 22 | Clock signal for SPI communication |
|  | SDA | 23 | Data line (equivalent to MOSI for SPI displays) |
|  | RES | 21 | Reset pin (active low) |
|  | DC | 19 | Data/Command control pin |
|  | CS | 18 | Chip Select for LCD SPI |
|  |  |  | Backlight control (often active high) |
| **SIM800L** | RST | 4 | Reset pin for SIM800L |
|  | RXD | 17 | SIM800L RXD ← connect to ESP32 TX (TX2 port) |
|  | TXD | 16 | SIM800L TXD → connect to ESP32 RX (RX2 port) |
| **MP3-TF-16P v3.0** | RX | 14 | MP3 module RX ← connect to ESP32 TX (TX1 port) |
|  | TX | 13 | MP3 module TX → connect to ESP32 RX (RX1 port) |
| **ESP32-CAM** | RX | Nc | ESP32-CAM RX ← connect to ESP32 TX (TX0 port) |
|  | TX | Nc | ESP32-CAM TX → connect to ESP32 RX (RX0 port) |

**Additional Considerations**

* **Logic Levels:**  
  Although the ESP32 (and ESP32-CAM module) typically run at 3.3V, double-check that all connected modules (LCD, SIM800L, MP3 module) are compatible with the chosen voltage levels. You might need level shifters if any peripheral expects a different voltage level.
* **UART Ports:**  
  The SIM800L, MP3 module, and ESP32-CAM are using separate UART/serial lines. Ensure that the ESP32’s available UART peripherals (TX0/RX0, TX1/RX1, TX2/RX2) are correctly configured and not in conflict with other functions (like programming/debugging).
* **SPI vs. Serial:**  
  The LCD uses SPI signals (even though its pin names are labeled SDA/SCL, they function as MOSI/SCK in this context). Ensure your display’s library is set up for SPI mode and that the pins match the assigned ones.

Review your board’s schematic and documentation to confirm the pin mapping and required voltage levels. If you have any further questions or need clarification on integrating the power supply/regulation, feel free to ask!