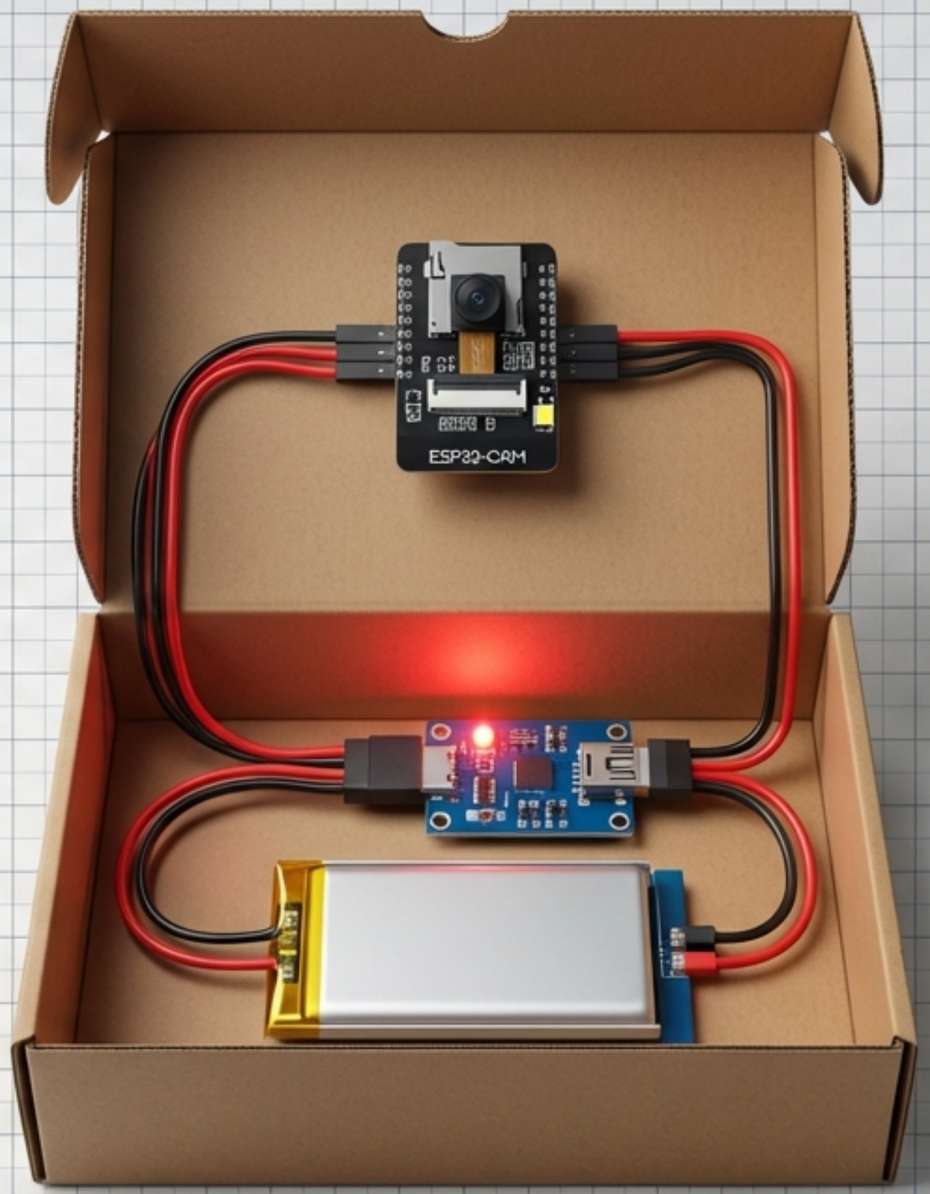
I was Created To Create.

The journey of building a portable, wireless video streaming system with ESP32-CAM.

A mini-project by Ashwin Krishna.

The Vision: A Low-Cost, Portable, Wireless Streaming Camera.

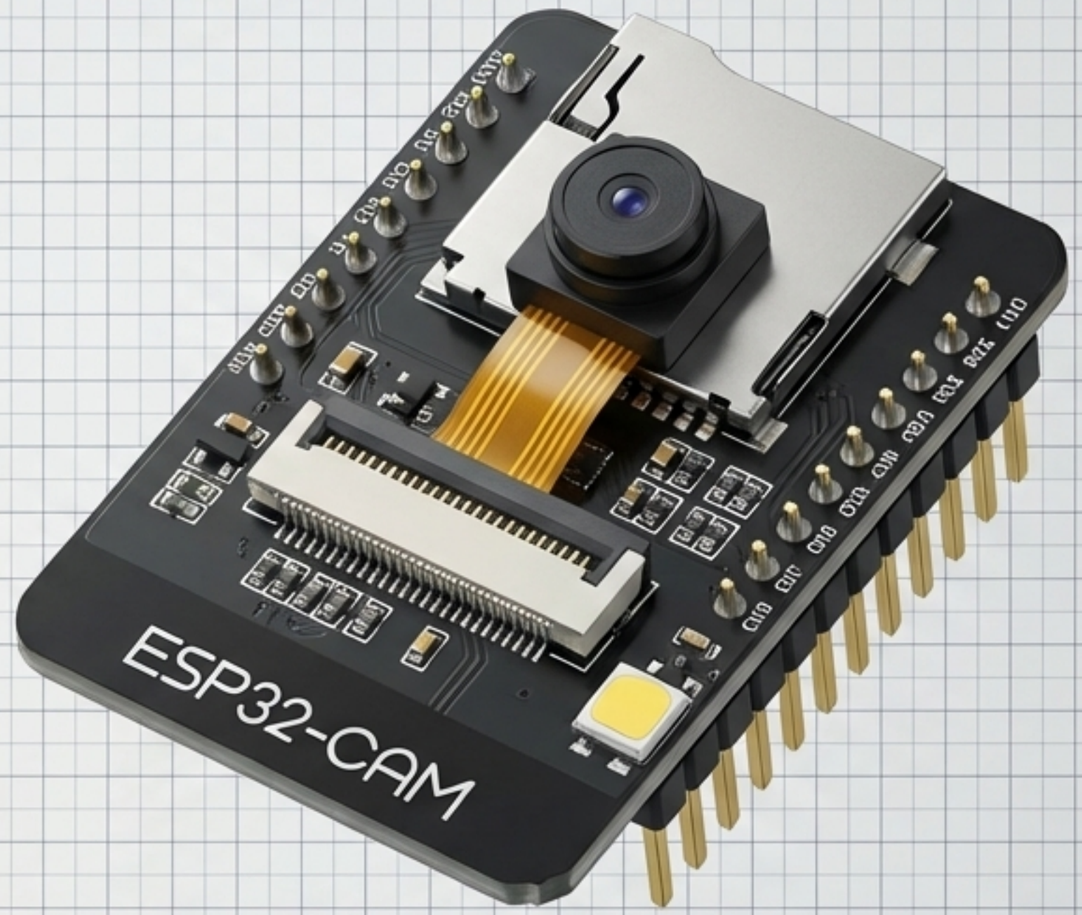
- This project documents the creation of a self-contained, battery-powered video streaming system built around the ESP32-CAM module.
- The primary goal was to build a portable device capable of capturing video and transmitting it wirelessly to a laptop or server for real-time monitoring.
- Conceived as a 3rd-semester engineering mini-project, it demonstrates a practical application of IoT principles on a minimal budget.



The Core Component: The ESP32-CAM Module.

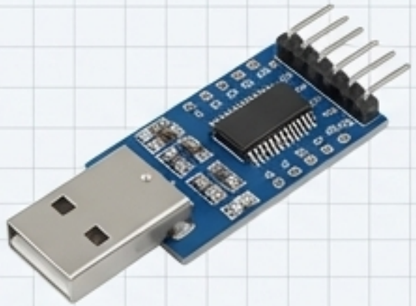
The ESP32-CAM is a compact, low-cost development board that integrates a powerful ESP32 microcontroller with a camera module. Its versatility makes it ideal for IoT projects requiring image processing and computer vision.

- **Powerful Processor:** Dual-core ESP32 for processing data.
- **Integrated Camera:** Typically the OV2640, for real-time image and video capture.
- **Wireless Connectivity:** Built-in Wi-Fi and Bluetooth for streaming and communication.
- **Expandability:** GPIO pins for interfacing with other sensors and actuators.
- **Storage:** Onboard microSD card slot.

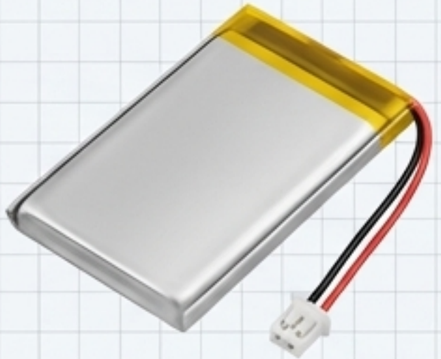


The Blueprint: Assembling the Toolkit for under ₹600.

ESP32-CAM Module



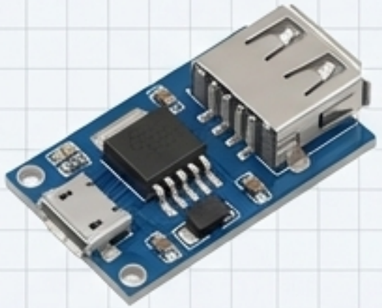
FTDI Programmer



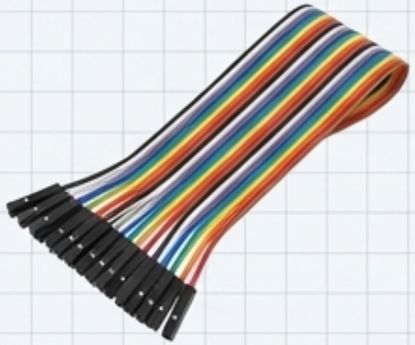
1500mAh Li-ion Battery

Total Project Cost: ~₹600

*(Note: All components sourced from robocraze.in)



5V Step-Up Voltage Regulator



Jumper Wires

ON/OFF Switch

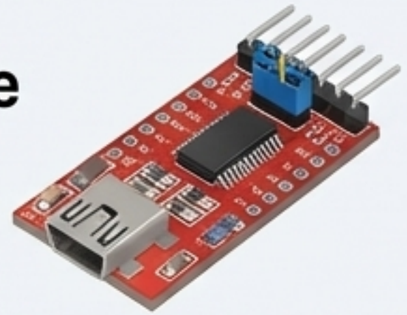
Phase I: Breathing Life into the Code with Arduino IDE.

1. Install Environment

The primary environment for programming the board.

2. Connect Hardware

Use the FTDI programmer to create a communication link.



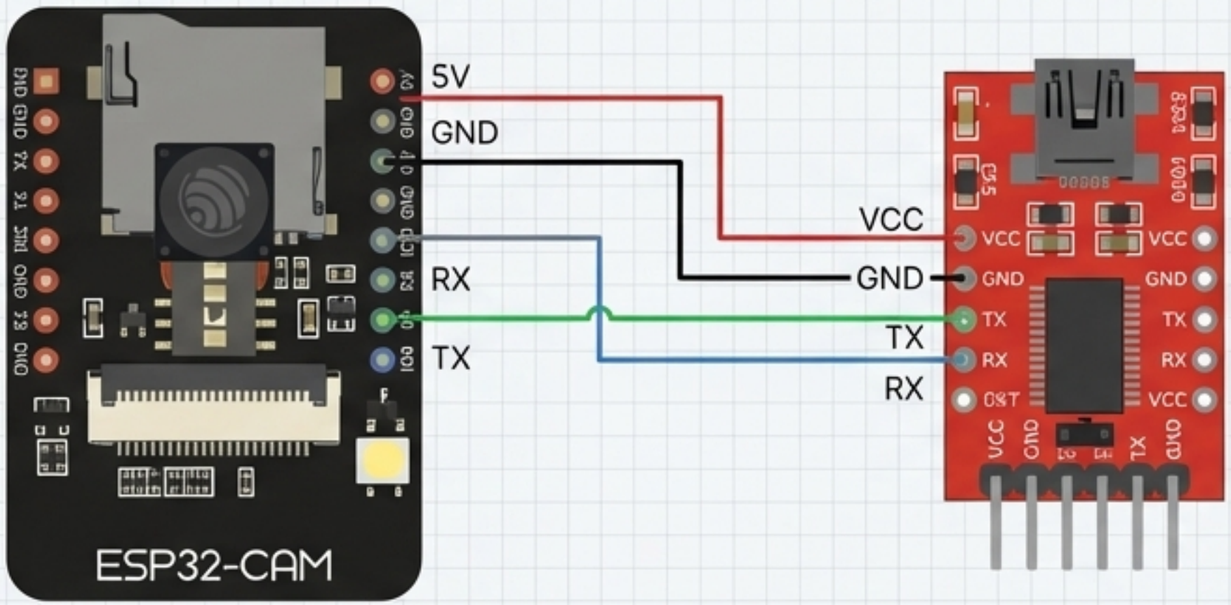
3. Load & Configure Code

Use the pre-built “CameraWebServer” sketch as a starting point.

4. Upload to Board

Flash the microcontroller with the configured configured software.

The Programming Interface: Connecting the FTDI Programmer

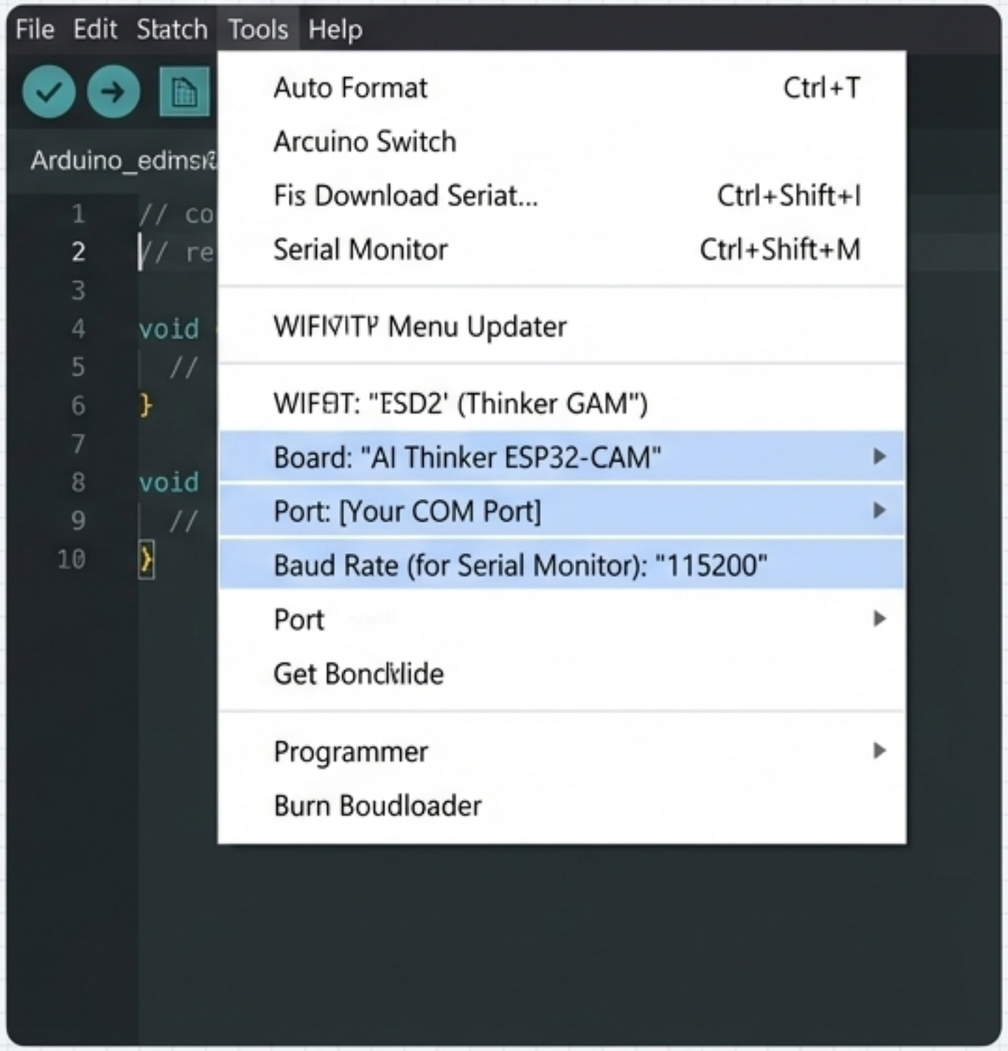


Pinout List:

- FTDI 5V → ESP32-CAM 5V
- FTDI GND → ESP32-CAM GND
- FTDI TX → ESP32-CAM RX
- FTDI RX → ESP32-CAM TX

Critical Note: To enter programming mode, connect pin IO0 to GND before uploading.

Software Settings



Configuring the Stream: Wi-Fi Credentials and Camera Model.

In the 'CameraWebServer' example sketch, two main changes are required: uncommenting the correct camera model and adding your network credentials.

```
// Select camera model
//#define CAMERA_MODEL_WROVER_KIT
//#define CAMERA_MODEL_ESP_EYE
#define CAMERA_MODEL_AI_THINKER

// Enter your Wi-Fi credentials
const char* ssid = "YOUR_WIFI_SSID";
const char* password = "YOUR_WIFI_PASSWORD";
```

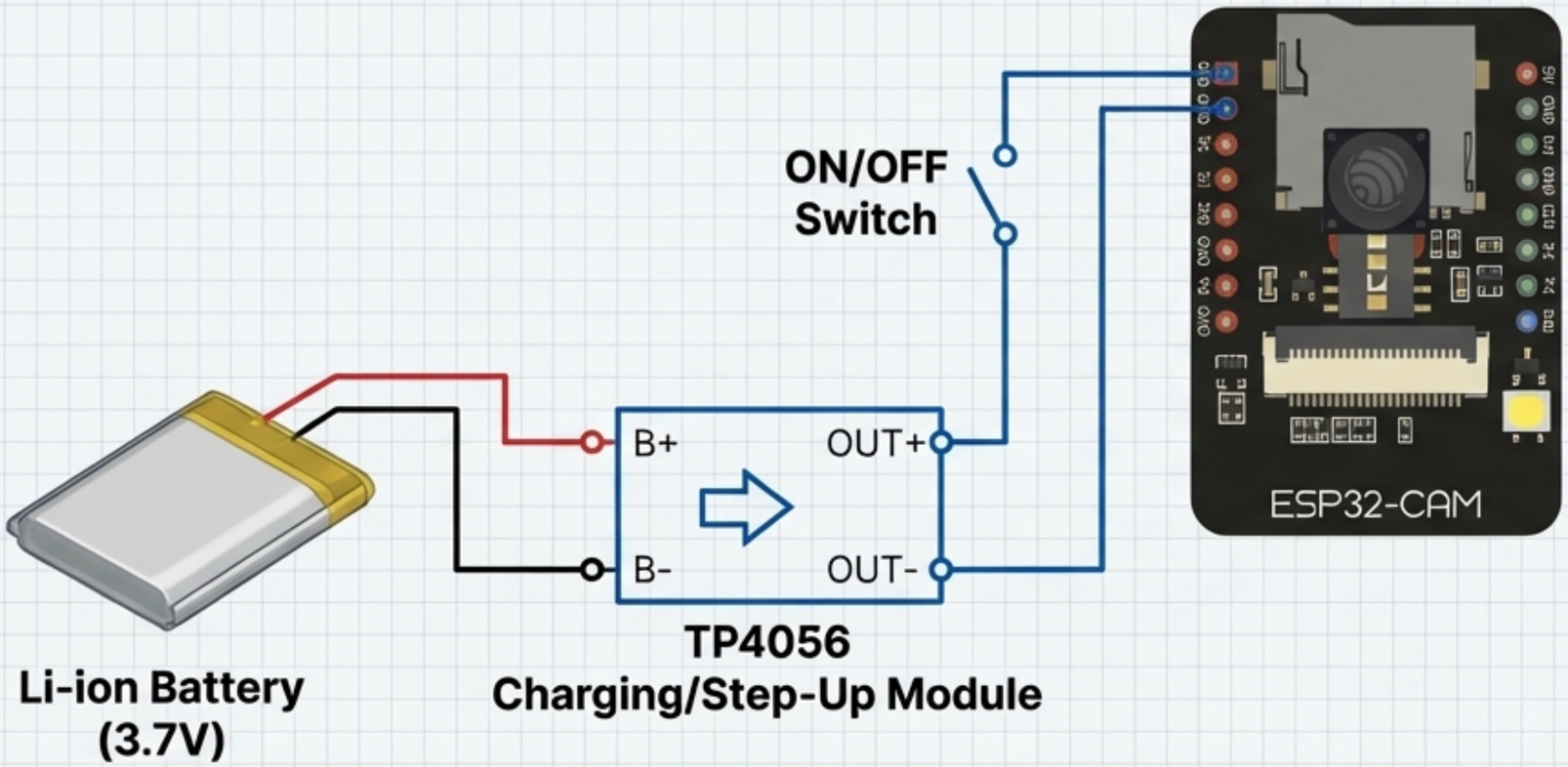
Uncomment this line.

Add your network details here.

Phase II: The Physical Assembly and Power Circuit.

With the code uploaded, the FTDI programmer is no longer needed. The next step is to build the permanent, battery-powered circuit.

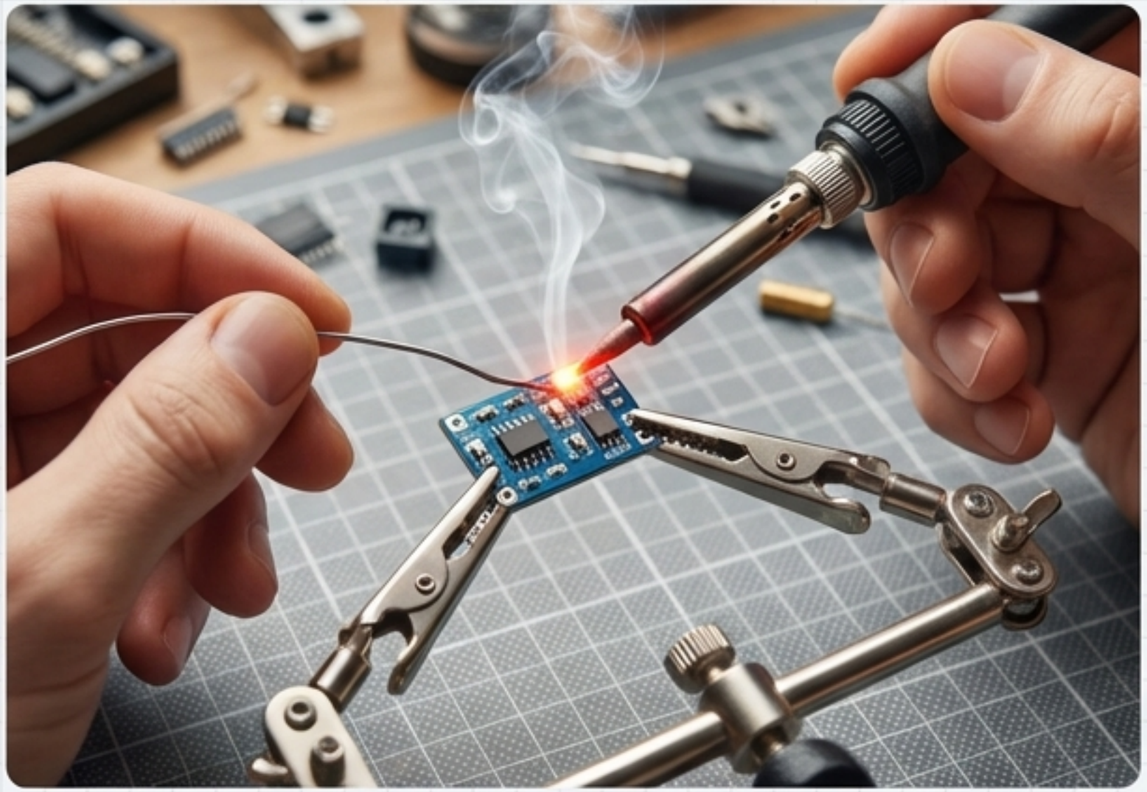
Note: Soldering is required for secure and lasting connections.

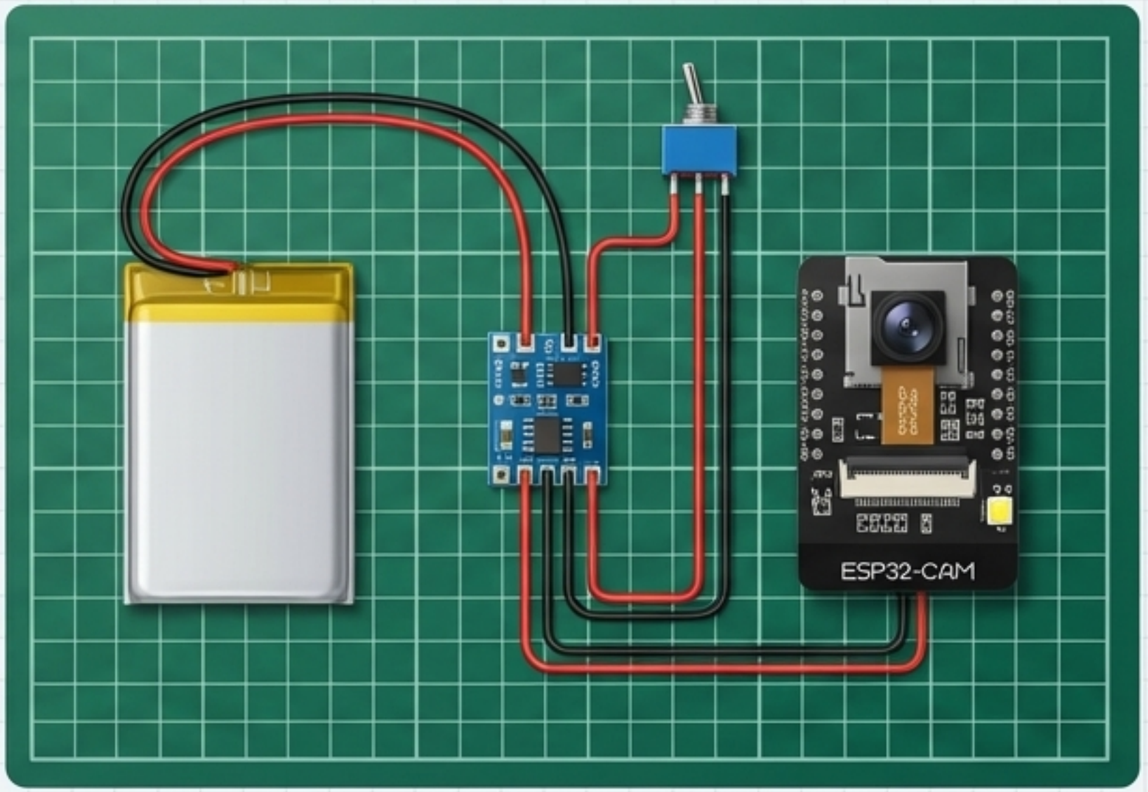


Connection Flow:

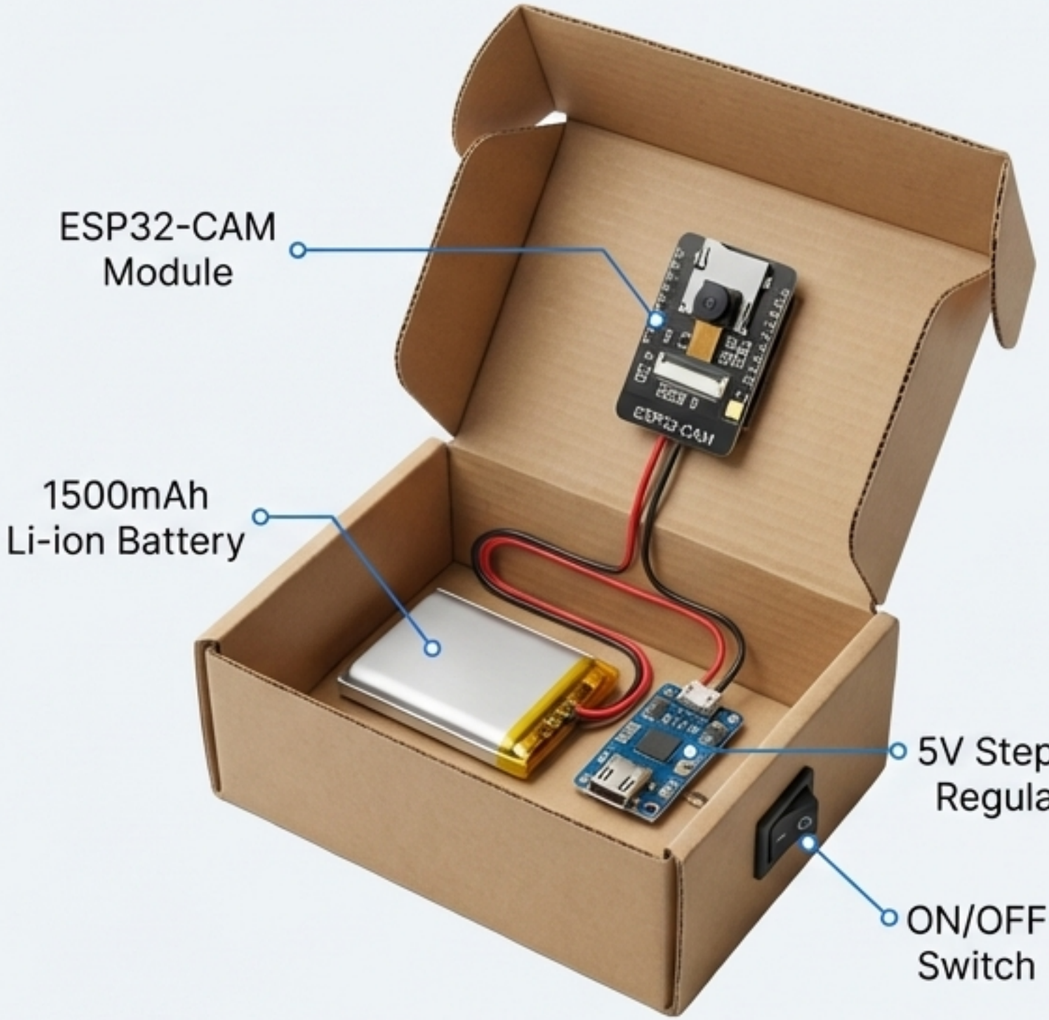
- **Battery** (3.7V) → **Input** of Step-Up Regulator
- **Output** (5V) of Step-Up Regulator → **VIN/GND** of ESP32-CAM
- An ON/OFF switch is integrated between the regulator and the ESP32-CAM for power control.

From Components to Circuit: The BId in Progress.





The Unveiling: A Fully Enclosed, Portable Streaming Device.



Assembly Note: Components are secured inside the container using double-sided tape.

System Workflow: From Power-On to Live Stream.

1. Power On: Flip the switch. The device boots up.

2. Auto-Connect: The ESP32-CAM automatically connects to the pre-programmed Wi-Fi network.

3. Get IP Address: Open the Arduino IDE Serial Monitor (at 115200 baud) and press the ESP32-CAM's reset button. The device's IP address will be printed. (e.g., `http://192.168.1.107`).

4. Start Streaming: Copy the IP address and paste it into any web browser on the same Wi-Fi network to view the live video stream.

Key Requirement: The ESP32-CAM and the viewing device (laptop/phone) must be connected to the same Wi-Fi network.

Project Vitals: Key Specifications and Performance

BUDGET

~₹600

(approx. $7-8 USD)

POWER

Source: 1500mAh 3.7V Li-ion Battery

Operating Voltage: Stable 5V (via Step-Up Regulator)

BATTERY LIFE

1-2 Hours

(Continuous video streaming)

From Mini-Project to Real-World Solution: Potential Applications.

Home Security Systems

- Use as a portable, discrete security camera.
- Integrate face detection to identify visitors or grant access to authorized individuals.
- Set up alerts for unrecognized faces.

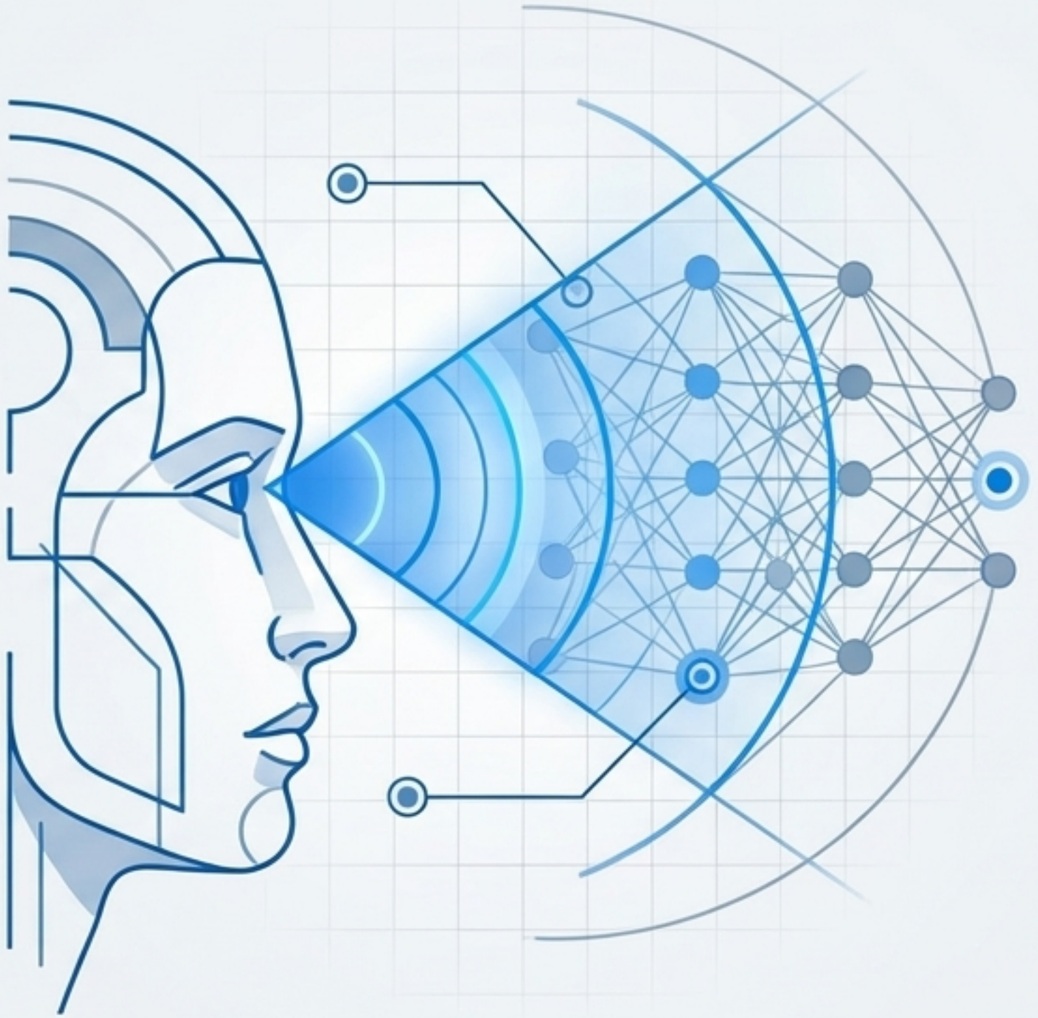
Automated Attendance Tracking

- Deploy in classrooms or offices to log attendance automatically.
- Eliminates manual intervention and streamlines record-keeping.
- Can be adapted for event check-ins.

The Next Chapter: Future Enhancements & Capabilities.

The current system provides a powerful foundation for more advanced machine learning applications. Future upgrades could include:

- **Emotion Recognition:** Analyze facial expressions to gauge reactions, useful for research or interactive installations.
- **Age and Gender Detection:** Gather demographic data to provide tailored services or experiences.
- **Mask Detection:** Implement a system to monitor compliance with health and safety protocols in public spaces.



A Complete Blueprint for a Portable Vision System.

This project successfully demonstrates the creation of a portable, wireless, and affordable video streaming device powered by the ESP32-CAM. The journey from individual components to a functional, enclosed system showcases the potential of modern microcontrollers for rapid prototyping and practical IoT solutions.

> "I was Created To Create."

Ashwin Krishna
LinkedIn: https://www.linkedin.com/in/ashxris/