To interface all three devices—ESP32-CAM, MFRC522 RFID reader, and a load cell (via HX711, typically)—into a unified \*\*penguin weighbridge system\*\*, here's a strategy:

---

### 🧠 \*\*System Architecture Overview\*\*

#### \*\*Devices:\*\*

1. \*\*ESP32-CAM\*\*

- Captures and optionally uploads an image.

2. \*\*ESP32 + HX711 + Load Cell\*\*

- Measures weight.

3. \*\*MFRC522 (on ESP32)\*\*

- Reads RFID tag for penguin ID.

---

### 🧩 \*\*Integration Plan (3 Options)\*\*

#### \*\*🅰️ All-in-One ESP32 (Best for Compact Design)\*\*

- Use a \*\*single ESP32 with sufficient GPIOs\*\*.

- Connect:

- HX711 to two GPIOs.

- MFRC522 via SPI (MOSI/MISO/SCK/SS).

- OV2640 camera via ESP32-CAM (needs specific GPIOs).

- ❗Challenge: GPIO conflicts, especially with camera + SPI + HX711.

- ✅ Possible, but needs careful pin mapping and possibly software SPI for RFID or HX711.

#### \*\*🅱️ Dual Microcontroller Setup (ESP32-CAM + ESP32)\*\*

- \*\*ESP32-CAM\*\* handles:

- Taking photo.

- Receiving weight + RFID info via serial or WiFi from second ESP32.

- \*\*Second ESP32\*\* handles:

- MFRC522 (SPI)

- HX711 (weight)

- Sends data to ESP32-CAM.

➡️ Communication between ESP32s via:

- \*\*UART\*\* (Serial)

- \*\*WiFi (HTTP or UDP packet)\*\*

- \*\*I2C Master/Slave\*\* (less common)

#### \*\*🅲️ Master ESP32 with External SPI Multiplexer\*\*

- Use an \*\*SPI bus multiplexer\*\* (like 74HC4051) to avoid SPI conflicts.

- Advanced, but lets you keep a clean single-board design.

---

### 🛠️ Recommended Implementation (Option B)

\*\*Why?\*\*: ESP32-CAM has limited free GPIOs. Using a second ESP32 offloads sensors and avoids conflicts.

#### ⛓️ Connections:

- ESP32-CAM ←(WiFi or UART)→ ESP32 (sensors)

- ESP32 reads:

- HX711 load cell (e.g., pins 4, 5)

- MFRC522 via SPI (e.g., 18, 19, 23, 5)

- Sends weight and RFID tag when penguin steps on bridge.

---

### 📸 Workflow Example

1. Load cell reads weight → if weight > threshold → trigger event.

2. RFID is scanned for tag → get penguin ID.

3. ESP32 sends ID + weight to ESP32-CAM.

4. ESP32-CAM takes photo and stores/sends it with metadata (timestamp, ID, weight).

---

### 📦 Optional: Data Upload

- ESP32-CAM sends image and data via:

- HTTP POST to cloud

- Save on SD card (if available)

- Local web server

---

Would you like a diagram and code structure to guide your implementation?

**Step 1: Establish ESP-NOW Communication**

* ✅ **Goal:** Get ESP32 and ESP32-CAM talking using ESP-NOW.
* 🔧 **Actions:**
  + Set up one ESP32 as **sender** and the ESP32-CAM as **receiver**.
  + Exchange MAC addresses.
  + Verify successful message transmission using Serial Monitor logs.

#### **Step 2: Implement a Trigger Event Over ESP-NOW**

* ✅ **Goal:** Trigger an action on the ESP32-CAM from the sender ESP32.
* 🔧 **Actions:**
  + On sender: send a basic command like "CAPTURE" over ESP-NOW.
  + On ESP32-CAM: parse incoming message and, if it matches "CAPTURE", call a dummy function (e.g., Serial.println("Triggered!")) to simulate image capture.

#### **Step 3: Capture and Save Image on SD Card**

* ✅ **Goal:** Capture a photo and store it when triggered via ESP-NOW.
* 🔧 **Actions:**
  + Initialize the SD card on the ESP32-CAM.
  + Replace dummy function with real camera capture code.
  + Save image with timestamp or unique filename on SD card.

#### **Step 4: Send Simulated RFID + Weight Data**

* ✅ **Goal:** Simulate final integration.
* 🔧 **Actions:**
  + Format and send a string like: "ID1234,2.7kg" from ESP32.
  + On ESP32-CAM, parse the string.
  + Optionally, embed this info into the image filename or log it.

Step 1: Connect Esp32CAM To Esp using ESP NOW

Step 2: Try creating a trigger event sent from ESP32 TO ESP32 CAM to take image using espnow

Step 3: Try to capture image using ESP32 Cam and store it on SD Card from trigger event

Step 4: Send String message from ESP32 TO ESP32 Cam like “Hi ID 123 Alive” Simulating weight + RFID Data

Step 5: Stop inputs from main system to send image to database? No loose some data from the main system

Step 5: If data send fail try after 10s? assuming internet speed of 10mps

Extract the data

Heavy EM Interface Reduces range

Setup RFID

Recive RFID Data

Extract Weight

Send Signal to ESP32 Cam

Capture image

Send data to server