Trường Đại học Bách Khoa – Đại học Đà Nẵng
Khoa Điện tử - Viễn thông

ĐỀ TÀI:
HỆ THỐNG TỰ ĐỘNG PHÂN LOẠI TÁO DỰA TRÊN MÀU SẮC

Sinh viên thực hiện :

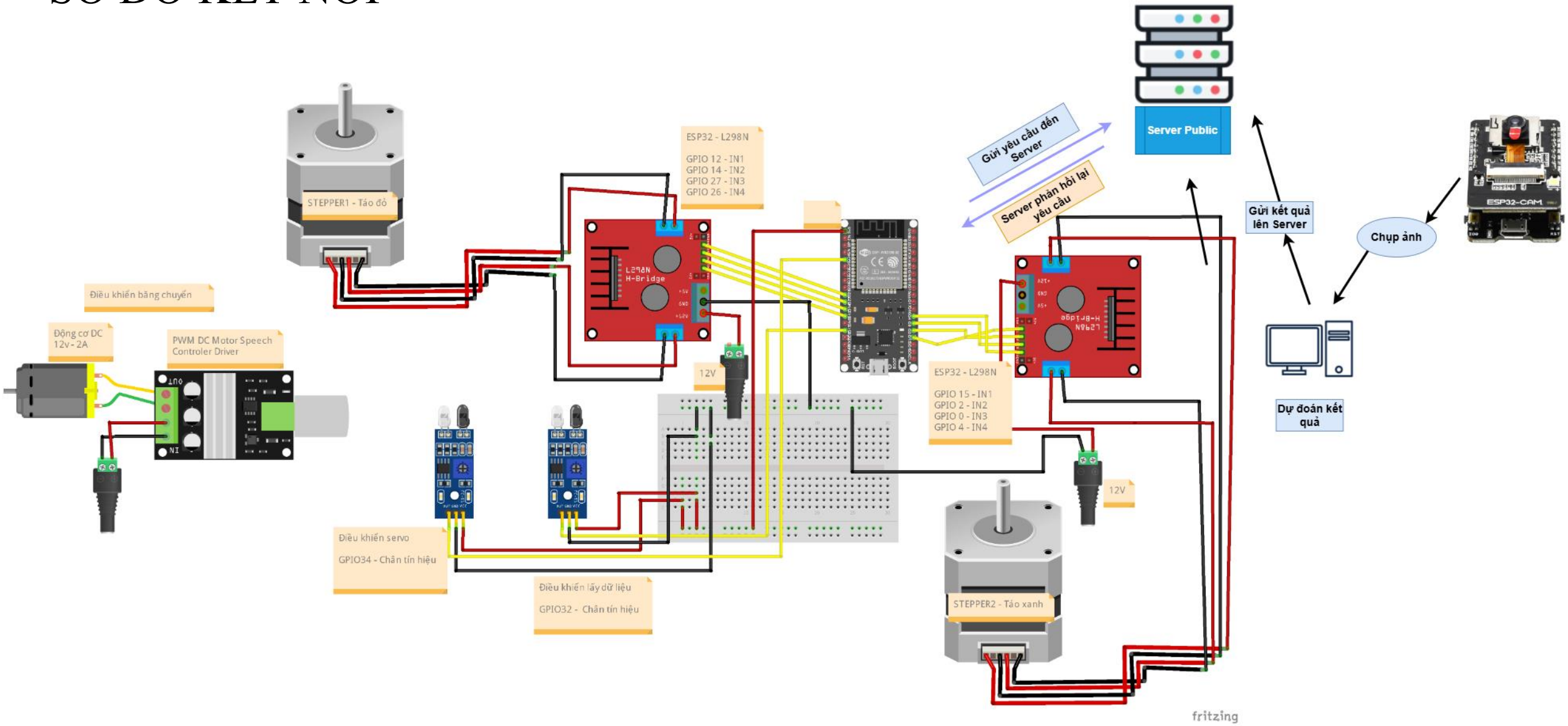
1. Hồ Đức Vũ 20KTMT2
2. Nguyễn Minh Phương 20KTMT1
3. Nguyễn Văn Vĩnh Quang 20KTMT1
4. Lê Tuấn Nhật 20KTMT2

Nhóm HP : 20.44

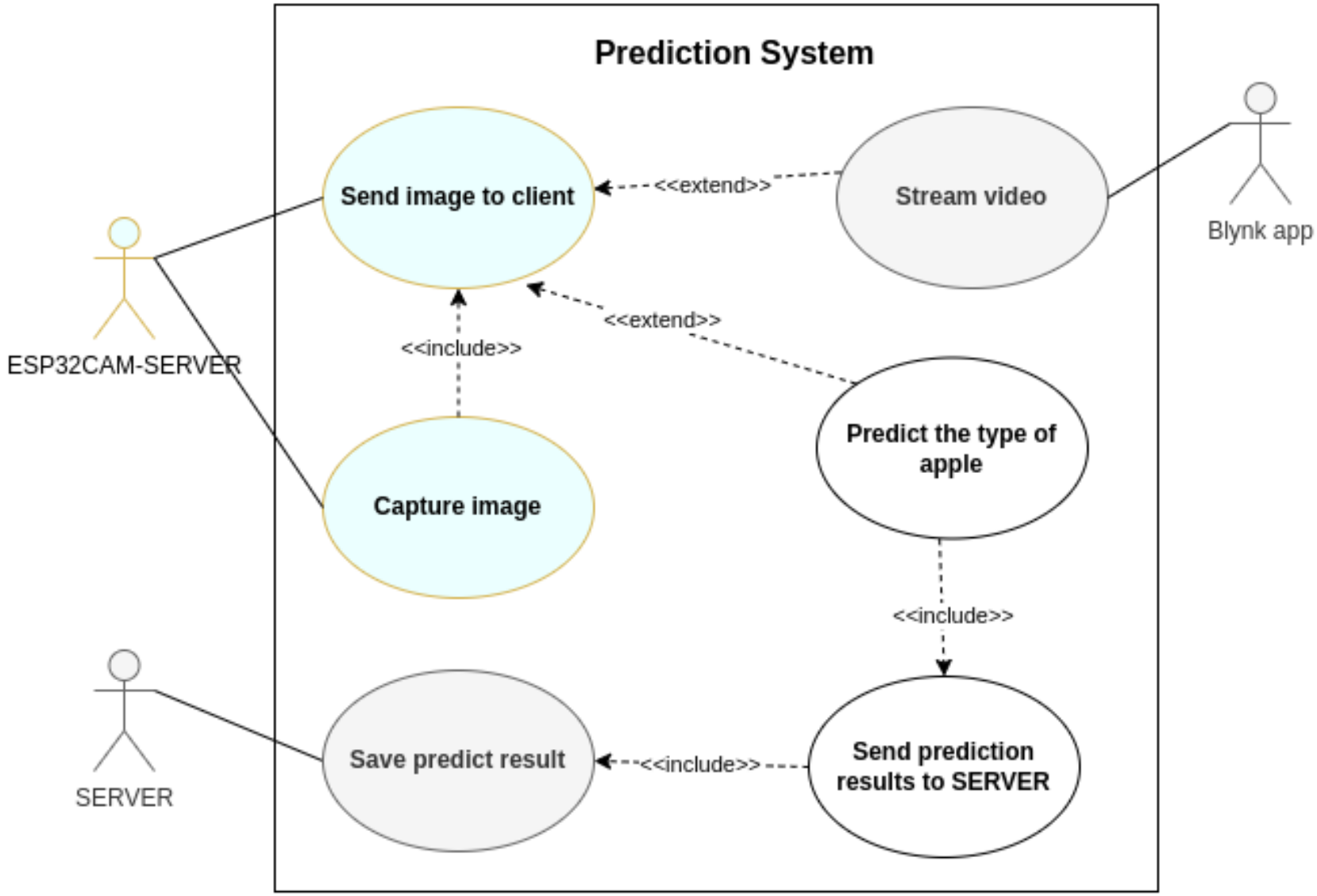
GVHD :

ThS. HỒ VIẾT VIỆT

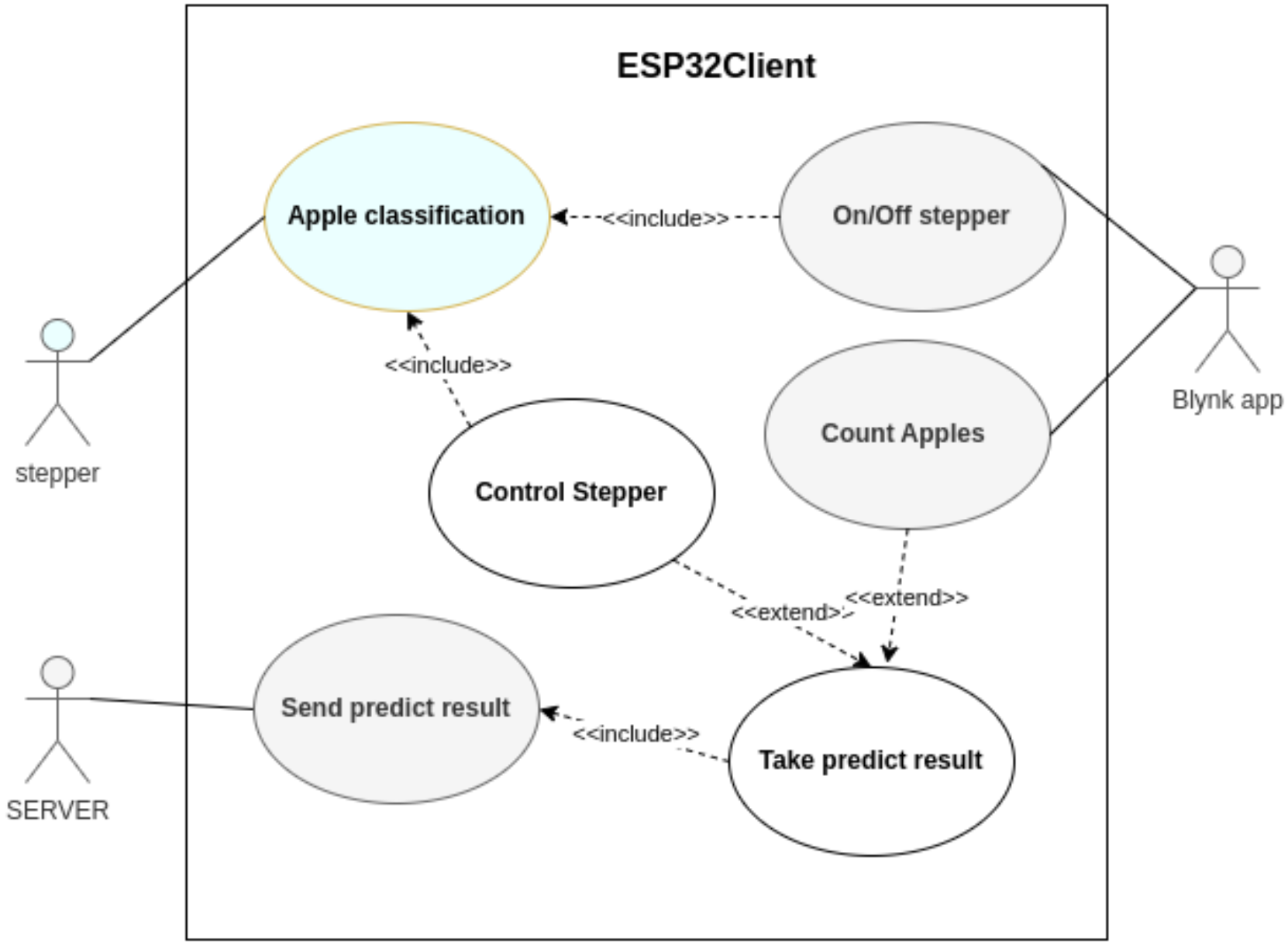
SƠ ĐỒ KẾT NỐI

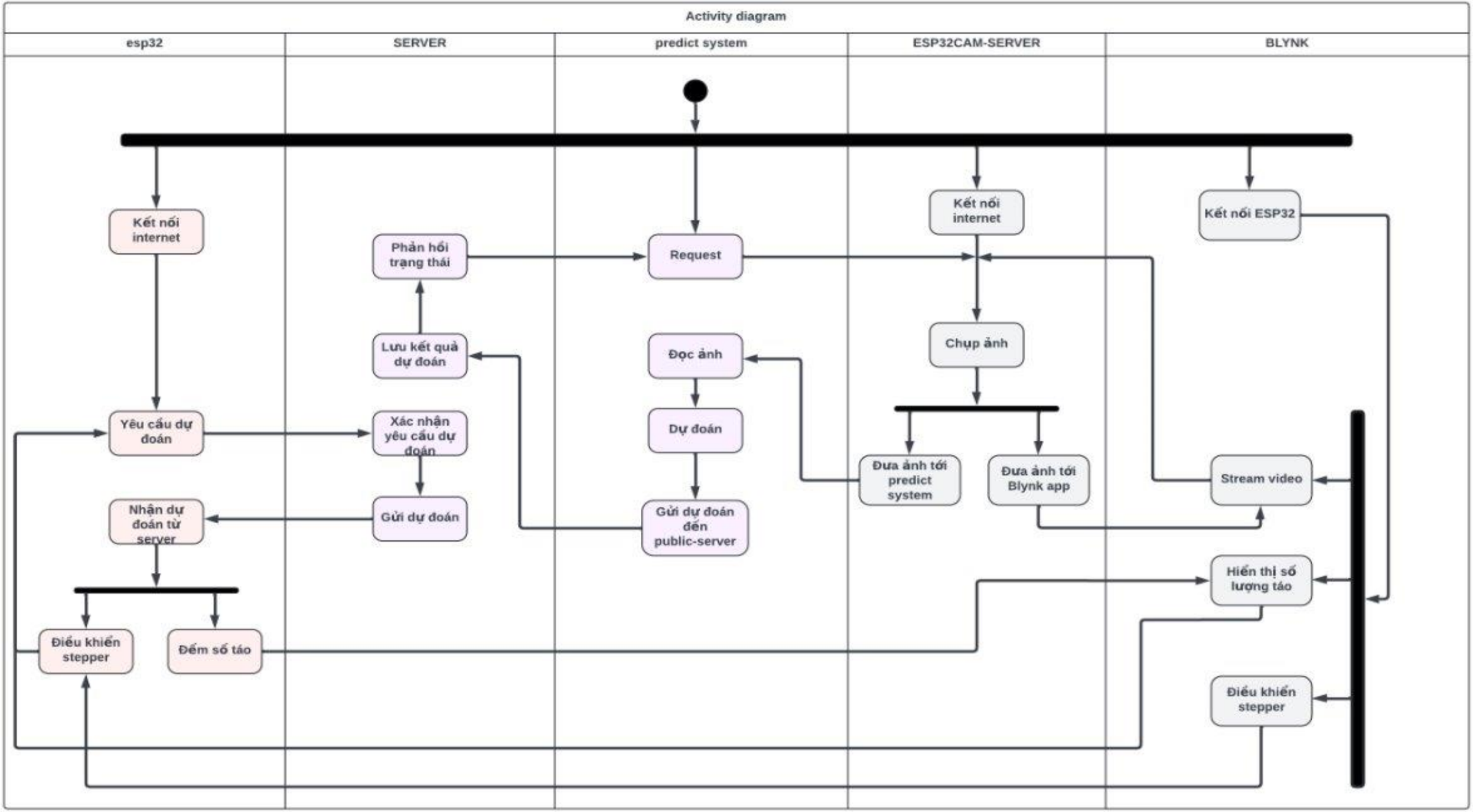

STEPPER1 - Táo đỏ
ESP32 - L298N
GPIO 12 - IN1
GPIO 14 - IN2
GPIO 27 - IN3
GPIO 26 - IN4
L298N H-Bridge
Điều khiển băng chuyền
Động cơ DC 12v - 2A
PWM DC Motor Speech Controler Driver
12V
Điều khiển servo
GPIO34 - Chân tín hiệu
Điều khiển lấy dữ liệu
GPIO32 - Chân tín hiệu
ESP32 - L298N
GPIO 15 - IN1
GPIO 2 - IN2
GPIO 0 - IN3
GPIO 4 - IN4
12V
STEPPER2 - Táo xanh
Gửi yêu cầu đến Server
Server phản hồi lại yêu cầu
Server Public
Gửi kết quả lên Server
Chụp ảnh
ESP32-CAM
Dự đoán kết quả
fritzing

Use case diagram



Use case diagram



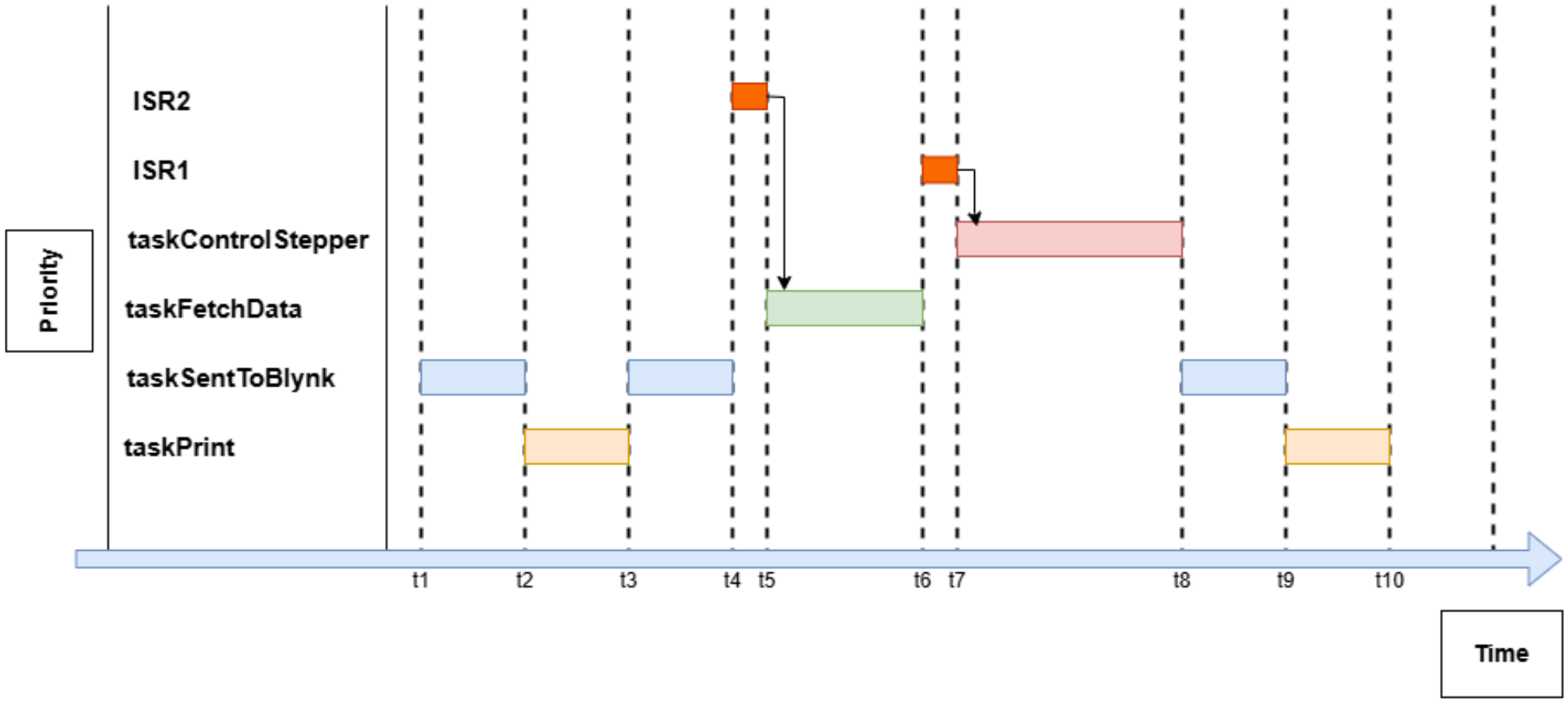

Activity diagram
esp32
SERVER
predict system
ESP32CAM-SERVER
BLYNK
Kết nối internet
Phản hồi trạng thái
Request
Kết nối internet
Kết nối ESP32
Lưu kết quả dự đoán
Đọc ảnh
Chụp ảnh
Yêu cầu dự đoán
Xác nhận yêu cầu dự đoán
Dự đoán
Đưa ảnh tới predict system
Đưa ảnh tới Blynk app
Stream video
Nhận dự đoán từ server
Gửi dự đoán
Gửi dự đoán đến public-server
Hiển thị số lượng táo
Điều khiển stepper
Đếm số táo
Điều khiển stepper

```cpp
WebServer captureServer(80);   // Server for predict system
WebServer streamServer(81);    // Server for streaming on Blynk app
....
void captureTask(void *pvParameters) {
  captureServer.on("/cam-hi.jpg", handleJpgHi);
  captureServer.begin();
  for (;;) {
    captureServer.handleClient();
  }
}

void streamTask(void *pvParameters) {
  streamServer.on("/monitor", []() {
    handleMjpeg();
  });
  streamServer.begin();
  for (;;) {
    streamServer.handleClient();
  }
}
....
void setup() {
    ...
    Serial.print("Capture server: http://");
    Serial.print(WiFi.localIP());
    Serial.println("/cam-hi.jpg");

    Serial.print("Stream server: http://");
    Serial.print(WiFi.localIP());
    Serial.println(":81/monitor");

    xTaskCreatePinnedToCore(captureTask, "Capture Task", 4096, NULL, 1, NULL, 0);
        Core 0
    xTaskCreatePinnedToCore(streamTask, "Stream Task", 4096, NULL, 1, NULL, 1);
        Core 1
}
```

```python
# Python
def predict():
    while True:
        try:
            # Request image from the camera feed
            img_resp = urllib.request.urlopen(cam_url)
            # Read image
            imgnp = np.array(bytearray(img_resp.read()), dtype=np.uint8)
            im = cv2.imdecode(imgnp, -1)
            # Try predict
            pre = model.make_predict(im)
            print(f"Predict: {pre}")
            # Prepare the data to send
            data = {
                "prediction": pre,
                "status": "200"
            }
            # Make a POST request to send the prediction to the SERVER
            response = requests.post(f'{url_server}', json=data, timeout=10)
            if response.status_code == 200:
                print("Prediction successfully posted!")
            else:
                print(f"Error posting prediction: {response.text}")
        except Exception as e:
            print(f"Error in predict function: {e}")
```

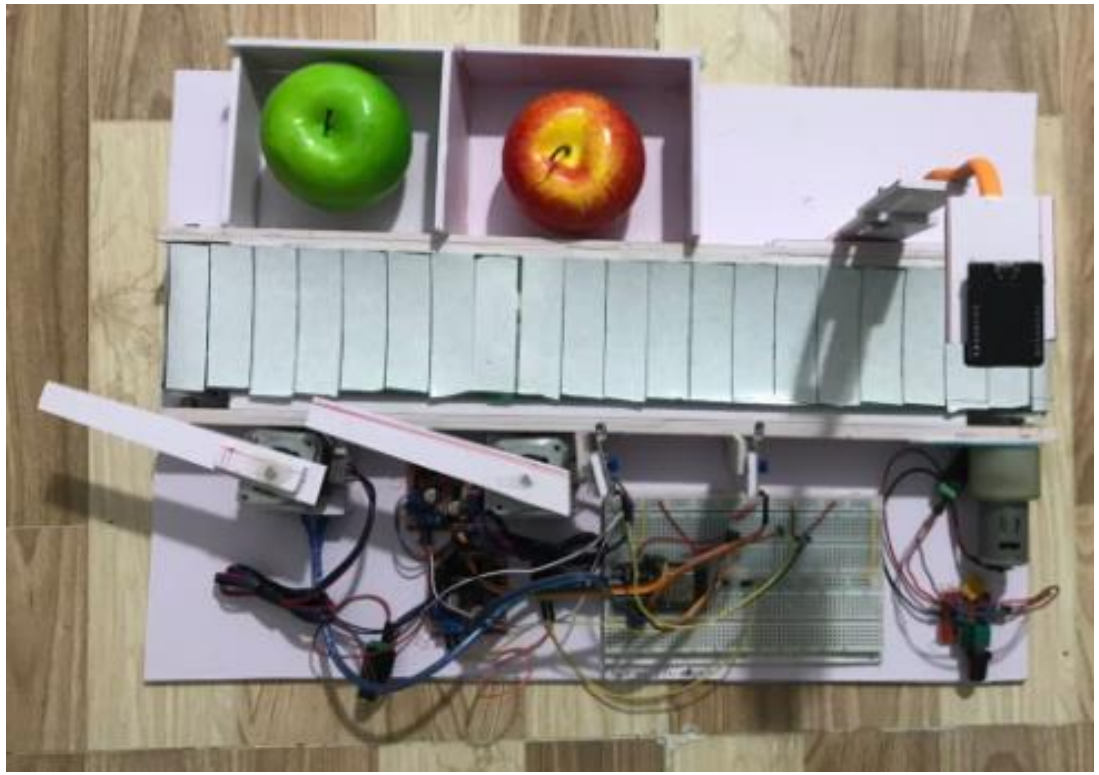
```python
@app.post('/upload', status_code=status.HTTP_200_OK)
async def predict(data: PredictionRequest):
    try:
        prediction = data.prediction
        status = data.status
        # Monitoring the prediction
        logging.info(f"Received prediction: {prediction} with status: {status}")
        # Store the prediction in the pred_list
        pred_list.append(data.dict())

        return {"message": "Prediction received successfully", "prediction":
            prediction, "status": status}
    except Exception as e:
        logging.error(f"Error in prediction: {e}")
        raise HTTPException(status_code=500, detail="Error in prediction")
```

Timing diagram

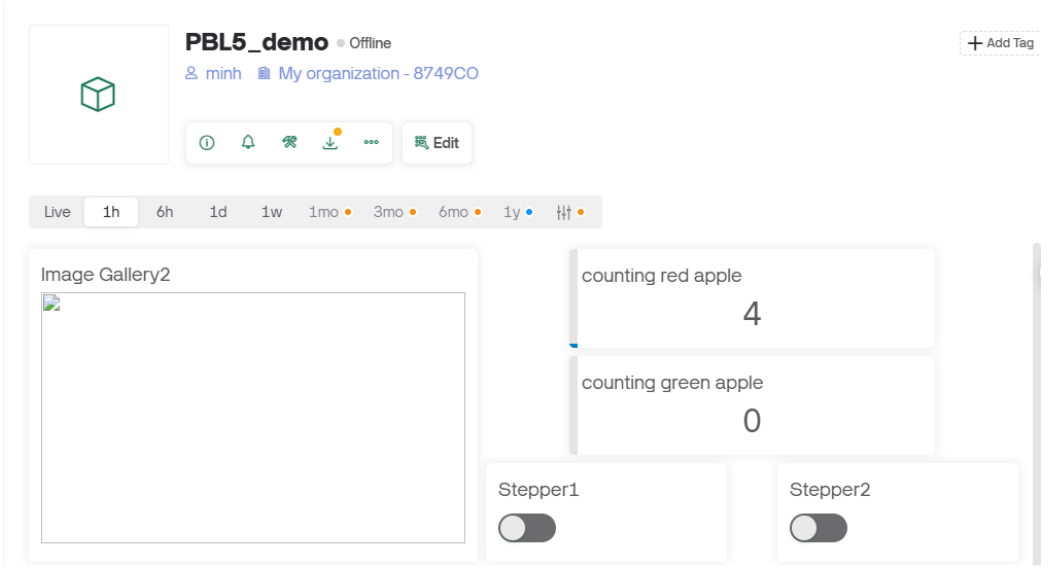


esp32

Kết quả



Mô hình băng chuyền



Giao diện Blynk

DEMO


THANK YOU