A picture containing text, sign

Description automatically generatedLogo

Description automatically generated**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA - ĐẠI HỌC ĐÀ NẴNG**

**KHOA CƠ KHÍ**

**🙠🙠🕮🙢🙢**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: KỸ THUẬT CƠ – ĐIỆN TỬ**



**ĐỀ TÀI: THIẾT KẾ HỆ THỐNG PHÂN LOẠI RÁC**

**TÁI CHẾ SỬ DỤNG KỸ THUẬT HỌC SÂU**

|  |  |
| --- | --- |
| **SVTH:** | **Đặng Nhật Trường, 101180209** |
|  | **Nguyễn Thanh Hạ, 101180167** |
| **GVHD:** | **TS. Đặng Phước Vinh** |
|  |  |

**Đà Nẵng, 12/2022**

MỤC LỤC

[nhiệm vụ đồ án tốt nghiệp 5](#_Toc120354997)

[I. Tên đề tài: 5](#_Toc120354998)

[THIẾT KẾ HỆ THỐNG PHÂN LOẠI RÁC TÁI CHẾ SỬ DỤNG KỸ THUẬT HỌC SÂU 5](#_Toc120354999)

[II. Nhiệm vụ: 5](#_Toc120355000)

[III. Ngày giao nhiệm vụ: 5](#_Toc120355001)

[IV. Ngày hoàn thành: 5](#_Toc120355002)

[V. Cán bộ hướng dẫn: 5](#_Toc120355003)

[lịch trình thực hiện đồ án tốt nghiệp 6](#_Toc120355004)

[lời cam đoan 7](#_Toc120355005)

[LỜI NÓI ĐẦU 8](#_Toc120355006)

[CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 9](#_Toc120355007)

[1.1. Giới thiệu về phương pháp thực hiện: 9](#_Toc120355008)

[1.2. Giới thiệu về SCADA: 9](#_Toc120355009)

[1.3. Sơ đồ khối của hệ thống: 10](#_Toc120355010)

[1.4. Giới thiệu về công nghệ học sâu: 10](#_Toc120355011)

[CHƯƠNG 2: GIỚI THIỆU CÁC THIẾT BỊ trong HỆ THỐNG và phần mềm lập trình 11](#_Toc120355012)

[2.1. PLC Mitsubishi FX1N 60MT: 11](#_Toc120355013)

[2.1.1. Tổng quan về PLC: 11](#_Toc120355014)

[2.1.2. Đặc tính của PLC Mitsubishi FX1N 60MT: 12](#_Toc120355015)

[2.2. Cáp lập trình PLC Mitsubishi USB-SC09-FX: 13](#_Toc120355016)

[2.3. Cảm Biến Quang E3F-DS30C4 30CM NPN 6-36V: 14](#_Toc120355017)

[2.4. Camera: 15](#_Toc120355018)

[2.5. Tủ điện: 15](#_Toc120355019)

[2.5.1. Relay 24VDC – 220VAC: 16](#_Toc120355020)

[2.5.2. Cầu dao tự động – Aptomat 20A: 18](#_Toc120355021)

[2.6. Phần mềm GX Works2: 19](#_Toc120355022)

[2.7. Phần mềm Microsoft Visual Studio 2017: 19](#_Toc120355023)

[2.8. Phần mềm SQL Server 2014: 21](#_Toc120355024)

[2.9. Phần mềm KEPServer EX 6: 22](#_Toc120355025)

[chương 3: Thiết kế TỦ ĐIỆN, băng tải VÀ MÔ HÌNH hệ thống 23](#_Toc120355026)

[3.1. Thiết kế tủ điện: 23](#_Toc120355027)

[3.2. Thiết kế băng tải: 25](#_Toc120355028)

[3.2.1. Các phương án thiết kế băng tải: 25](#_Toc120355029)

[3.2.2. Chọn phương án thiết kế: 26](#_Toc120355030)

[3.2.3. Thiết kế băng tải: 26](#_Toc120355031)

[3.3. Mô hình hệ thống: 34](#_Toc120355032)

[chương 4: viết chương trình PLC, deep learning và giao diện, chức năng hệ thống SCADA 35](#_Toc120355033)

[4.1. Viết chương trình: 35](#_Toc120355034)

[4.1.1. Chương trình PLC: 35](#_Toc120355035)

[4.1.2. Chương trình C#: 40](#_Toc120355036)

[4.1.3. Chương trình python: 43](#_Toc120355037)

[4.2. Chức năng hệ thống SCADA: 49](#_Toc120355038)

[4.2.1. Quản lý người dùng: 49](#_Toc120355039)

[4.2.2. Giám sát: 49](#_Toc120355040)

[4.2.3. Điều khiển: 49](#_Toc120355041)

[4.2.4. Trạng thái cảm biến: 49](#_Toc120355042)

[4.2.5. Watchdog: 49](#_Toc120355043)

[4.2.6. Quản trị cơ sở dữ liệu: 49](#_Toc120355044)

[4.2.7. Xuất báo cáo file PDF, EXCEL, WORD: 49](#_Toc120355045)

[4.3. Phân loại bằng công nghệ xử lý ảnh: 49](#_Toc120355046)

[chương 5: kết luận 51](#_Toc120355047)

[5.1. Giới hạn đề tài: 51](#_Toc120355048)

[5.2. Hướng phát triển: 51](#_Toc120355049)

[tài liệu tham khảo 52](#_Toc120355050)

[Hình 1 - 1: Sơ đồ khối của hệ thống 10](#_Toc116852445)

[Hình 2 - 1: Thành phần cấu tạo PLC (Nguồn: Global Automation) 12](#_Toc116852446)

[Hình 2 - 2: Hình ảnh PLC Mitsubishi FX1N 60MT 14](#_Toc116852447)

[Hình 2 - 3: Cáp lập trình PLC Mitsubishi USB-SC09-FX 15](#_Toc116852448)

[Hình 2 - 4: Cảm biến tiệm cận hồng ngoại E3F-DS30P1 PNP (Nguồn: dientudat.com) 15](#_Toc116852449)

[Hình 2 - 5: Hình ảnh camera (Nguồn: Internet) 16](#_Toc116852450)

[Hình 2 - 6: Hình ảnh tủ điện điều khiển hệ thống 17](#_Toc116852451)

[Hình 2 - 7: Relay trung gian 24VDC – 220VAC (Nguồn: Internet) 18](#_Toc116852452)

[Hình 2 - 8: Sơ đồ mạch của relay trung gian 24VDC - 220VAC 18](#_Toc116852453)

[Hình 2 - 9: Hình ảnh cầu dao tự động Panasonic 20A 19](#_Toc116852454)

[Hình 2 - 10: Giao diện lập trình của phần mềm GX Work 2 20](#_Toc116852455)

[Hình 2 - 11: Giao diện lập trình Designer trên Visual Studio 2017 21](#_Toc116852456)

[Hình 2 - 12: Giao diện lập trình Code trên Visual Studio 2017 21](#_Toc116852457)

[Hình 2 - 13: Giao diện làm việc của phần mềm SQL Server 22](#_Toc116852458)

[Hình 2 - 14: Giao diện làm việc của phần mềm KEPServer EX 6 23](#_Toc116852459)

[Hình 3 - 1: Hình ảnh tủ điện với các nút nhấn, công tắc 24](#_Toc116852460)

[Hình 3 - 2: Hình ảnh kết cấu bên trong tủ điện - hệ thống SCADA 25](#_Toc116852461)

[Hình 3 - 3: Sơ đồ mạch điện của tủ điện 25](#_Toc116852462)

# nhiệm vụ đồ án tốt nghiệp

Họ tên: Đặng Nhật Trường MSSV: 101180209

Nguyễn Thanh Hạ MSSV: 101180167

Chuyên ngành: Kỹ thuật cơ – điện tử Mã ngành: …

Hệ đào tạo: Đại học chính quy Mã hệ: …

Khóa: 2018 Lớp: 18CDT1

## Tên đề tài:

## THIẾT KẾ HỆ THỐNG PHÂN LOẠI RÁC TÁI CHẾ SỬ DỤNG KỸ THUẬT HỌC SÂU

## Nhiệm vụ:

…

## Ngày giao nhiệm vụ:

Tháng 9 năm 2022.

## Ngày hoàn thành:

Tháng 12 năm 2022.

## Cán bộ hướng dẫn:

**TS**. **Đặng Phước Vinh**

# lịch trình thực hiện đồ án tốt nghiệp

Họ tên sinh viên 1: Đặng Nhật Trường MSSV: 101180209

Lớp: 18CDT1

Họ tên sinh viên 2: Nguyễn Thanh Hạ MSSV: 101180167

Lớp: 18CDT1

**Tên đề tài:** Thiết kế hệ thống phân loại rác tái chế sử dụng kỹ thuật học sâu

|  |  |  |
| --- | --- | --- |
| Tuần 1 + 2 | ĐN. Trường | Nghiên cứu, lên lịch trình, dựng CSDL SQL. |
|  | NT. Hạ | Nghiên cứu, lên lịch trình. |
| Tuần 3 + 4 | ĐN. Trường | Thiết kế bảng vẽ điện (eCAD). |
|  | NT. Hạ | Thiết kế cơ cấu cơ khí (CAD). |
| Tuần 5 + 6 | ĐN. Trường | Thi công phần điện (tủ điện + mạch điện + cảm biến). |
|  | NT. Hạ | Thi công phần cơ khí (băng tải). |
| Tuần 7 + 8 | ĐN. Trường | Xây dựng giao diện hệ thống SCADA (C#) + code PLC. |
|  | NT. Hạ | Chuẩn bị dữ liệu để training. |
| Tuần 9 + 10 + 11 | ĐN. Trường | Hoàn thiện chức năng + dữ liệu cho hệ thống SCADA (C#) + code PLC. |
|  | NT. Hạ | Xây dựng module training. |
| Tuần 12 + 13 | ĐN. Trường | Vận hành máy, kiểm tra phần điện, phần mềm lập trình PLC và SCADA (C#) |
|  | NT. Hạ | Vận hành máy, kiểm tra phần cơ khí + AI. |
| Tuần 14 + 15 | ĐN. Trường | Hoàn thiện thuyết minh, viết báo. |
|  | NT. Hạ | Hoàn thiện dữ liệu, tính toán phần cơ khí và AI. |
| Tuần 16 |  | Hoàn thành. |

# lời cam đoan

Đề tài “Thiết kế hệ thống phân loại rác tái chế sử dụng kỹ thuật học sâu” là nhóm tôi tự thực hiện dựa vào việc tham khảo thực trạng môi trường ở một số bài báo và vài video về ý tưởng chọn đề tài. Bên cạnh đó còn có tài liệu hướng dẫn thiết kế băng tải trên internet và module AI trên github. Ngoài ra không sao chép từ tài liệu khác hay công trình sẵn có trước đó, kể cả ứng dụng SCADA trong hệ thống.

Nhóm thực hiện đề tài

**Đặng Nhật Trường & Nguyễn Thanh Hạ**

# LỜI NÓI ĐẦU

Hiện nay, vấn đề tái chế rác thải ở nước ta đang được mọi người quan tâm. Rác thải chưa được tiến hành tái chế một cách quy mô và cũng chưa có công nghệ đạt yêu cầu. Một trong những giai đoạn quan trọng ban đầu để giải quyết vấn đề đó là phân loại rác tái chế.

Đồ án này trình bày nghiên cứu của nhóm tác giả về hệ thống phân loại rác tái chế sử dụng kỹ thuật học sâu, giải quyết được vấn đề cốt lõi đầu tiên của việc tái chế rác thải là phân loại chúng.

Việc chôn lấp rác gây ô nhiễm cả tài nguyên nước và đất. “Rác hiện nay chưa được tiến hành tái chế và cũng chưa có công nghệ đạt yêu cầu”, Bộ trưởng Bộ TN&MT nói.

*(Theo báo Nhân Dân)*

Áp dụng machine learning vào hệ thống, nhóm tác giả muốn tận dụng công nghệ 4.0 vào việc sản xuất để tăng nhanh hiệu suất, tăng mức độ tự động hóa trong một hệ thống máy móc.

Nhóm tác giả còn xây dựng nên giải pháp SCADA mà trọng tâm là ứng dụng gọn nhẹ, dễ quản lý, người dùng dễ sử dụng, giao diện trực quan, thông số của máy rõ ràng, đầy đủ.

Ta chọn công cụ phần mềm thiết kế giao diện là Visual Studio (ngôn ngữ lập trình C#), phần mềm quản trị cơ sở dữ liệu là SQL Server, phần mềm truyền nhận dữ liệu trung gian là KEPServer EX, bộ điều khiển trung tâm là PLC Mitsubishi FX1N 60MT.

          Vì thời gian thực hiện đề tài có hạn nên đồ án này không trách khỏi những sai sót. Kính mong quý thầy cô và độc giả góp ý để hệ thống hoàn thiện hơn.

**Sinh viên thực hiện**

Đặng Nhật Trường, 101180209

Nguyễn Thanh Hạ, 101180167

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## Giới thiệu về phương pháp thực hiện:

*Mô tả tổng quát*:

PLC Mitsubishi FX1N 60MT sẽ là bộ điều khiển trung tâm, điều khiển hoạt động chính của máy. Người dùng có thể điều khiển máy trực tiếp trên tủ điện hoặc điều khiển, giám sát và thu thập dữ liệu trên hệ thống trên máy tính thông qua giao diện SCADA.

*Tóm tắt hướng thực hiện đề tài:*

+ Thiết kế băng tải phân loại sản phẩm.

+ Lập trình PLC Mitsubishi FX1N 60MT để tự động hóa nhà kính và giao tiếp với máy tính (C#).

+ Dùng phần mềm trung gian KEPServer EX để giao tiếp giữa PLC và giao diện SCADA (C#).

+ Lập trình winform trên máy tính bằng ngôn ngữ C# để thiết kế hệ thống SCADA.

+ Ứng dụng hệ quản trị cơ sở dữ liệu SQL Server của Microsoft để lưu trữ dữ liệu lâu dài.

+ Ứng dụng kỹ thuật học sâu để phân loại sản phẩm tái chế.

+ Dùng chân PWM tốc độ cao Y0, Y1 trên PLC để điều khiển động cơ servo đóng vai trò là cần gạt sản phẩm.

+ Dùng nguồn tổ ong 24V 5A để cấp nguồn cho ngõ ra của PLC, kích các relay điều khiển tải trong hệ thống.

## Giới thiệu về SCADA:

SCADA (Supervisory Control And Data Acquisition) là hệ thống điều khiển, giám sát và thu thập dữ liệu của một hệ thống nhằm hỗ trợ con người tương tác từ xa.

Cấu trúc của hệ thống SCADA bắt đầu bằng bộ điều khiển logic khả trình (PLC) hoặc thiết bị đầu cuối từ xa (RTU). PLC và RTU là các máy vi tính giao tiếp với các đối tượng như cảm biến, tải DC, tải AC, sau đó truyền thông tin của các đối tượng này đến máy tính bằng phần mềm SCADA. Phần mềm SCADA xử lý, phân phối và hiển thị dữ liệu cho người dùng.

*Hệ thống SCADA cho phép*:

* Kiểm soát các quy trình sản xuất tại mạng cục bộ hoặc từ xa.
* Theo dõi, thu thập và xử lý dữ liệu theo thời gian thực.
* Tương tác gián tiếp với các thiết bị như cảm biến, động cơ và nhiều thứ khác thông qua giao diện mà ta thiết lập.
* Ghi sự kiện, dữ liệu vào cơ sở dữ liệu.
* Xuất file báo cáo.
* Vẽ đồ thị, đưa ra cảnh báo.

## Sơ đồ khối của hệ thống:

Diagram

Description automatically generated

Hình 1 -1: Sơ đồ khối của hệ thống

## Giới thiệu về công nghệ học sâu:

Học sâu là một chức năng của trí tuệ nhân tạo (AI), bắt chước hoạt động của bộ não con người trong việc xử lí dữ liệu và tạo ra các mẫu để sử dụng cho việc ra quyết định.

Áp dụng trong hệ thống phân loại rác tái chế, nhóm em sẽ cho máy học hình ảnh của chai nhựa, các loại lon,… để máy tự động nhận dạng nhiều loại khác nhau và đưa ra quyết định phân loại đúng đắn.

# CHƯƠNG 2: GIỚI THIỆU CÁC THIẾT BỊ trong HỆ THỐNG và phần mềm lập trình

## PLC Mitsubishi FX1N 60MT:

### 2.1.1. Tổng quan về PLC:

PLC là từ viết tắt của Programmable Logic Controller (Bộ điều khiển logic khả trình). Khác với các bộ điều khiển thông thường chỉ có một thuật toán điều khiển nhất định, PLC có khả năng thay đổi thuật toán điều khiển tùy biến do người sử dụng viết thông qua ngôn ngữ lập trình. Do vậy, nó cho phép thực hiện linh hoạt tất cả các bài toán điều khiển.

Hiện nay có rất nhiều hãng sản xuất PLC như Siemens (Đức), Omron (Nhật Bản), Mitsubishi (Nhật Bản), Delta (Đài Loan),...

Ngôn ngữ lập trình phổ biến là LAD (Ladder logic - Dạng hình thang), FBD (Function Block Diagram - Khối chức năng), STL (Statement List - Liệt kê lệnh) và LAD là ngôn ngữ lập trình PLC đang được ưa chuộng nhất.

Tất cả các PLC hiện nay đều gồm có thành phần chính như sau:

* Bộ nhớ chương trình RAM, ROM.
* Một bộ vi xử lý trung tâm CPU, có vai trò xử lý các thuật toán.
* Các module tín hiệu vào/ra.

Diagram

Description automatically generated

Hình 2 - : Thành phần cấu tạo PLC (Nguồn: Global Automation)

*Nguyên lý hoạt động:*

Đầu tiên các tín hiệu từ các thiết bị ngoại vi (như các cảm biến) được đưa vào CPU thông qua module đầu vào. Sau khi nhận được tín hiệu đầu vào thì CPU sẽ xử lý và đưa các tín hiệu điều khiển qua module đầu ra xuất ra các thiết bị được điều khiển bên ngoài theo 1 chương trình đã được lập trình sẵn.

Một chu kỳ bao gồm đọc tín hiệu đầu vào, thực hiện chương trình, truyền thông nội, tự kiểm tra lỗi, gửi cập nhật tín hiệu đầu ra được gọi là 1 chu kỳ quét hay 1 vòng quét (Scan Cycle).

Thường thì việc thực hiện một vòng quét xảy ra trong thời gian rất ngắn (từ 1ms-100ms). Thời gian thực hiện vòng quét này phụ thuộc vào tốc độ xử lý lệnh của PLC, độ dài ngắn của chương trình, tốc độ giao tiếp giữa PLC và thiết bị ngoại vi.

*Ưu điểm của PLC:*

* Dễ dàng thay đổi chương trình theo ý muốn
* Thực hiện được các thuật toán phức tạp và độ chính xác cao.
* Cấu trúc PLC dạng module, cho phép dễ dàng thay thế, mở rộng đầu vào/ra, mở rộng chức năng khác
* Khả năng chống nhiễu tốt, hoàn toàn làm việc tin cậy trong môi trường công nghiệp.
* Giao tiếp được với các thiết bị thông minh khác như: Máy tính, nối mạng truyền thông với các thiết bị khác.

*Nhược điểm của PLC:*

* **Giá thành phần cứng cao, một số hãng phải mua thêm phần mềm để lập trình.**
* **Đòi hỏi người sử dụng phải có trình độ chuyên môn cao.**
* **Tốc độ xử lý không nhanh như vi điều khiển.**

### 2.1.2. Đặc tính của PLC Mitsubishi FX1N 60MT:

A close-up of a calculator on a book

Description automatically generated with low confidence

Hình 2 - : Hình ảnh PLC Mitsubishi FX1N 60MT

Thông số kỹ thuật:

* Nguồn vào: 110 - 220V AC.
* Ngõ vào: 36 ngõ vào DC.
* Ngõ ra: 24 ngõ ra transistor.
* Bộ nhớ chương trình: 8000 bước lệnh, sử dụng EEPROM.
* Không sử dụng pin.
* Chu kỳ xử lý lệnh: 0.55 – 0.7 µs.
* 1536 tiếp điểm phụ, 256 timers, 235 counters, 8000 thanh ghi.
* Ngõ ra xung tốc độ cao lên đến 100 kHz.
* Bộ đếm xung tốc độ cao (HSC):

1 phase: 6 ngõ vào với tần số tối đa 60 kHz

2 phase: 2 ngõ vào với tần số tối đa 30 kHz

* Board mở rộng giúp kết nối RS232C, RS485, RS422 dễ dàng.
* Board mở rộng có thể cung cấp 2 kênh ngõ vào analog và 1 kênh ngõ ra analog.
* Tích hợp 2 biến trở điều khiển có thể cài đặt bộ định thời.
* Hãng sản xuất: Mitsubishi – Nhật Bản.

Phần mềm lập trình: **GX-Works 2,** hoặc **GX-Developer,** hoặc **FX-PCS/WIN-E.**

## Cáp lập trình PLC Mitsubishi USB-SC09-FX:

A picture containing cable, connector, adapter

Description automatically generated

Hình 2 - : Cáp lập trình PLC Mitsubishi USB-SC09-FX

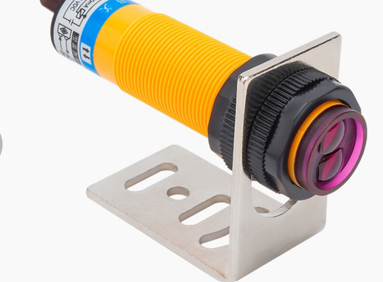
Chuẩn giao tiếp: RS422.

Giao tiếp với PC: cổng USB.

Chức năng: Download / upload chương trình / giao tiếp với máy tính cho các dòng PLC Mitsubishi FX.

Chiều dài cáp: 2.5 m.

## ****Cảm Biến Quang E3F-DS30C4 30CM NPN 6-36V:****



Hình 2 - : Cảm biến quang E3F-DS30C4 NPN

*Nguyên lý hoạt động của cảm biến:*

**Cảm Biến Quang E3F-DS30C4 NPN** dùng ánh sáng hồng ngoại để xác định có vật cản hay không.

Ngõ ra cảm biến dạng NPN khi không có vật cản ngõ ra ở mức cao, khi có vật cản ngõ ra ở mức thấp.

Ngoài ra, cảm biến có thể chỉnh khoảng cách nhận mong muốn thông qua biến trở được tích hợp trên mạch.

*Thông số kỹ thuật:*

* Nguồn điện cung cấp: DC 6~36V
* Khoảng cách phát hiện: 10~30cm.
* Có thể điều chỉnh khoảng cách qua biến trở.
* Dòng kích ngõ ra: 300mA.
* Ngõ ra dạng NPN.
* Chất liệu sản phẩm: nhựa.
* Sơ đồ chân:
  + Nâu: + 6~36V
  + Xanh: 0V
  + Đen: Tín hiệu ngõ ra
* Kích thước: 70 x 24mm

## Camera:

A picture containing diagram

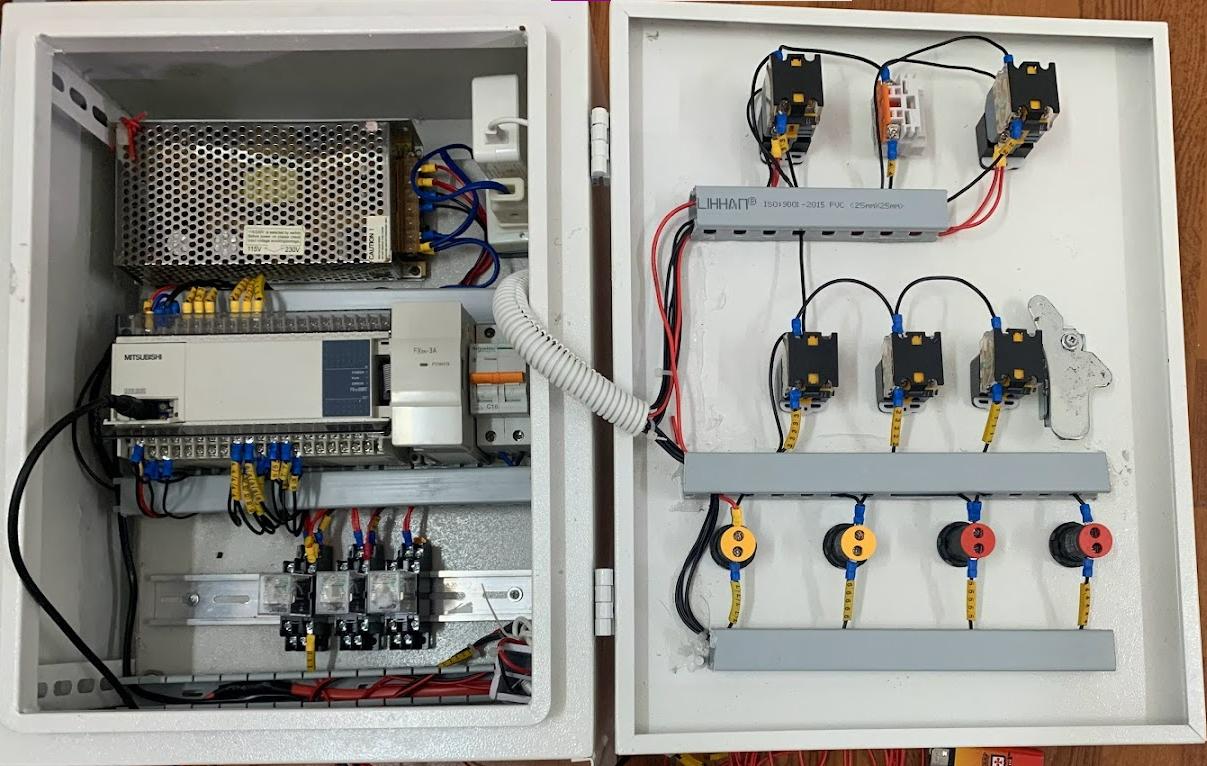
Description automatically generated

Hình 2 - : Hình ảnh camera (Nguồn: Internet)

*Thông số kỹ thuật:*

…updating…

## Tủ điện:



Hình 2 - : Hình ảnh tủ điện điều khiển hệ thống

* + Kích thước tủ điện: 45 x 35 x 18 cm.
  + Bộ điều khiển trung tâm: PLC Mitsubishi FX1N 60MT
  + Module mở rộng (Analog): FX0N-3A
  + Nguồn tổ ong: 24V 4.2A
  + Relay: 24VDC
  + Nút nhấn, công tắc, đèn báo: 24V

### Relay 24VDC – 220VAC:



Hình 2 - : Relay trung gian 24VDC – 220VAC (Nguồn: Internet)

Thông số:

– Điện áp cuộn hút (điện áp làm việc): 24VDC  
– Dòng cuộn hút tiêu thụ: 0.05A

– Số chân: 8 chân dẹp  
– Số lần đóng cắt: 100.000 lần  
– Có đèn báo đóng ngắt trên mỗi Relay  
– Bộ sản phẩm gồm: relay 8 chân dẹp + đế ra 8 chân  
– Dòng đóng cắt trên 1 tiếp điểm: 10A / 220vac – 10A / 28vdc  
– Số cặp tiếp điểm: 2 cặp NC (thường đóng), 2 CẶP NO (thường mở), loại tiếp điểm Silver alloy.

Chart, box and whisker chart

Description automatically generated

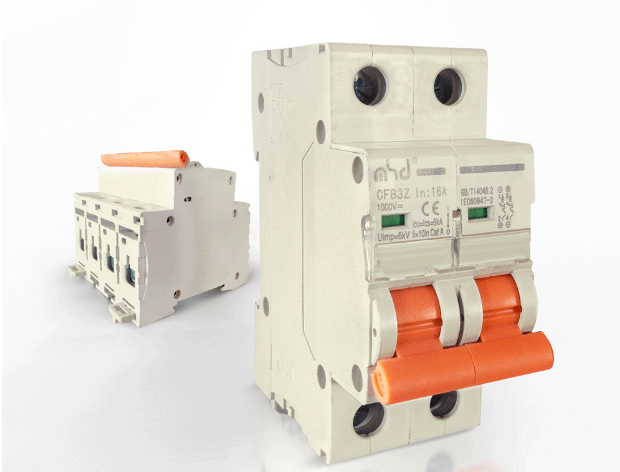
Hình 2 - : Sơ đồ mạch của relay trung gian 24VDC - 220VAC

Chúng có chức năng chuyển mạch tín hiệu điều khiển hoặc làm nhiệm vụ khuếch đại. Trong sơ đồ điều khiển, relay trung gian thông thường được lắp đặt ở vị trí trung gian. Nó nằm giữa những thiết bị điều khiển công suất nhỏ và những thiết bị có công suất lớn hơn. làm cầu nối giữa module điều khiển và các thiết bị đóng cắt động lực

Relay trung gian thường được sử dụng chịu tải trung gian cho các cảm biến, làm các mạch tự giữ, ON-OFF, với thiết kế có nhiều tiếp điểm nhờ đó mà ta có thể chỉ cần sử dụng một tín hiệu có thể điều khiển được nhiều thiết bị cùng một lúc.

Relay trung gian cũng là thiết bị bảo vệ cho các thiết bị báo mức ON-OFF, thay vì sủ dụng ngõ ra các cảm biến để kích van hay đèn… thì thông qua relay trung gian nó sẽ giúp cảm biến được cách ly hoàn toàn, nếu có sự cố quá tải thì chỉ hư các tiếp điểm relay trung gian.

### Cầu dao tự động – Aptomat 20A:



Hình 2 - : Hình ảnh aptomat chống giật 20A

**CB chống giật Panasonic** (hay còn gọi là Aptomat chống giật), ứng dụng giúp hệ thống tự động ngắt điện khi có chập điện, giật điện hay nguồn điện bị hở, rò rỉ ra ngoài, giúp bảo vệ an toàn điện.

*Thông số kĩ thuật của aptomat chống giật* 20A:

* Số cực: 1
* Dòng điện định mức: 20A
* Cấp điện áp 240VAC/ 415VAC
* Dòng cắt ngắn mạch định mức: 6kA
* Aptomat bảo vệ quá tải và ngắn mạch.
* Trọng lượng : 0.1kg

## Phần mềm GX Works2:

A screenshot of a computer

Description automatically generated with medium confidence

Hình 2 - : Giao diện lập trình của phần mềm GX Work 2

GX Works2 là phiên bản nâng cấp và thay thế cho phiên bản GX Developer bị hạn chế một số tính năng. Những cải tiến trong phiên bản này gồm:

* Giao diện được thiết kế lại trực quan, dễ lập trình hơn.
* Cập nhật thêm các thư viện module.
* Hỗ trợ thêm các ngôn ngữ lập trình FBD và SFC.
* Thao tác dễ dàng để tùy chỉnh các thông số.

## Phần mềm Microsoft Visual Studio 2017:

Chúng ta sẽ sử dụng phần mềm Visual Studio 2017 để lập trình thiết kế giao diện và chức năng của hệ thống SCADA bằng ngôn ngữ lập trình C#.

A picture containing text, monitor, indoor, screenshot

Description automatically generated

Hình 2 - : Giao diện lập trình Designer trên Visual Studio 2017

A screenshot of a computer

Description automatically generated with medium confidence

Hình 2 - : Giao diện lập trình Code trên Visual Studio 2017

Các tính năng của phần mềm Visual Studio:

* Hỗ trợ đa ngôn ngữ (HTML, CSS, C#, C/C++, JSON, JavaScript, Python, Node.js,…).
* Hỗ trợ Debug dễ dàng và mạnh mẽ (nhờ vào Breakpoint).
* Hỗ trợ kết nối với GitHub để đồng bộ các dự án.
* Kho tiện ích mở rộng phong phú (nhiều ngôn ngữ, thư viện).

## Phần mềm SQL Server 2014:

A computer screen capture

Description automatically generated with medium confidence

Hình 2 - : Giao diện làm việc của phần mềm SQL Server

SQL là ngôn ngữ truy vấn dữ liệu. Nó cung cấp tập lệnh phong phú cho các công việc lấy hay ghi dữ liệu:

* DDL (Data Definition Language): CREATE, ALTER, DROP, TRUNCATE
* DML (Data Manipulation Languge): INSERT, UPDATE, DELTE
* DCL (Data Control Language): GRANT, REVOKE

## Phần mềm KEPServer EX 6:

Table

Description automatically generated

Hình 2 - : Giao diện làm việc của phần mềm KEPServer EX 6

KEPServer EX là một tiêu chuẩn tương tác giao diện phần mềm cho phép trao đổi dữ liệu an toàn và đáng tin cậy giữa các chương trình Windows và các thiết bị phần cứng công nghiệp, bao gồm các loại PLC, các bộ điều khiển truyền thông Modbus, Profibus, Profinet,…

Nguyên lý hoạt động:

* KEPServer EX đóng vai trò trung gian thực hiện các kết nối dữ liệu từ PLC và Visual studio.
* KEPServer EX sẽ được cài đặt trong máy tính chạy SCADA, thông qua kết nối vật lý của máy tính với PLC, KEPServer lấy dữ liệu từ PLC và gửi dữ liệu đến Visual studio C# và ngược lại.

KEPServer EX hỗ trợ hầu hết các dòng PLC trên thị trường.

# chương 3: Thiết kế TỦ ĐIỆN, băng tải VÀ MÔ HÌNH hệ thống

## Thiết kế tủ điện:

Diagram

Description automatically generated

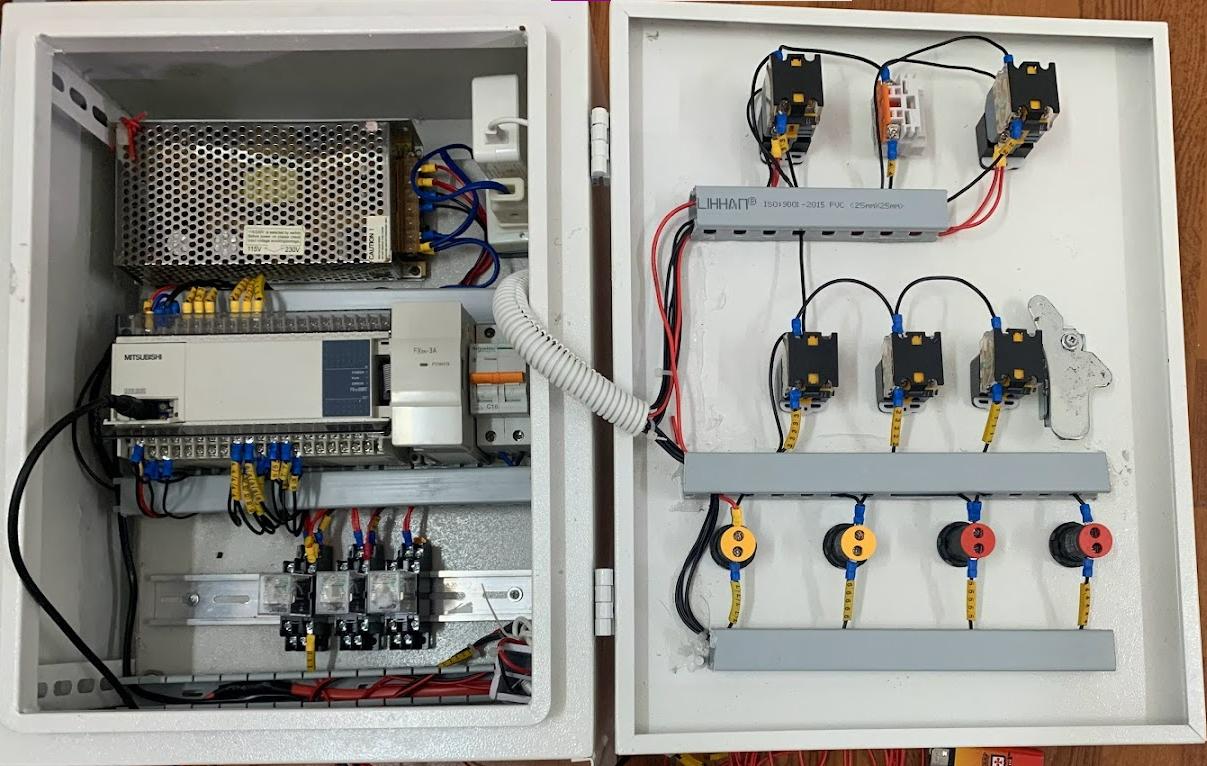
Hình 3 - : Bản vẽ Panel của tủ điện trước khi thi công thực tế

* Kích thước: 45 x 35 x 18 cm.
* Hệ thống nút nhấn gồm có:
* Nút nhấn Start
* Nút nhấn Stop
* Nút nhấn Emergency
* Công tắc đèn chiếu sáng, servo 1, servo 2
* Hệ thống LED báo trạng thái các phần của hệ thống

Diagram

Description automatically generated with low confidence

Hình 3 - : Bản vẽ kỹ thuật điện chi tiết của hệ thống



Hình 3 - : Hình ảnh kết cấu thực tế bên trong tủ điện

## Thiết kế băng tải:

### Các phương án thiết kế băng tải:

#### Phương án 1: Băng tải cao su

Băng tải cao su hiện nay được sử dụng rộng rãi trên thị trường. Thông thường băng tải cao su dùng để tải các loại vật liệu từ nặng cho đến nhẹ và các vật liệu đặc biệt (nhiệt độ cao, phân hóa học, …).

Loại băng tải này có các ưu nhược điểm sau đây:

**Ưu điểm:**

* Tải được các loại vật liệu có trọng lượng lớn: nhờ vào độ MPA cao, độ đàn hồi cao, số lớp bố nhiều và có độ dày lớn.
* Bền: Độ giãn theo chiều dọc ít
* Băng tải cao su có khả năng chống mài mòn cao
* Chịu được trong những môi trường có độ ẩm, nhiệt độ cao và hóa chất.
* Khó bị tách do các lớp bố và cao su được gắn kết rất chắc chắn.

Nhược điểm:

* Trong quá trình hoạt động, nếu băng tải chạy quá nhanh có thể dẫn đến tình trạng băng tải bị lệch trục và gây hại cho con lăn.
* Độ dốc của băng tải cao su thường nhỏ hơn so với các loại khác
* Không thể tải theo độ cao

#### Băng tải PVC:

B**ăng tải PVC**được rất nhiều doanh nghiệp ứng dụng trong các hoạt động sản xuất từ trước đến nay. Loại băng tải này thường được dùng để vận chuyển hàng hóa, đóng gói sản phẩm, phân loại sản phẩm.

Nó được ứng dụng trong rất nhiều ngành sản xuất như thực phẩm, lắp ráp linh kiện, sản xuất dược phẩm, y tế, may mặc, ...

Loại này có ưu điểm và nhược điểm:

Ưu điểm:

* Băng tải PVC trong sản xuất linh hoạt cung cấp cường độ cao, đặc tính co giãn thấp, hoàn toàn chống ẩm, chống gây mùi và nấm mốc.
* Dây băng tải hầu hết không thấm dầu mỡ và các hóa chất khác, đảm bảo độ an toàn khi chuyển tải hàng hóa, vật liệu.
* Độ bền kéo tốt, tuổi thọ dài, không bị dãn trong quá trình hoạt động.
* Hoạt động ổn định, năng suất vận hành cao, không gây tiêng ồn khi sử dụng.
* Nhiệt độ làm việc giao động từ -10ºC - 80ºC
* Có thể vận tải tốt trong các băng tải ngang, băng tải cong và cả băng tải nghiêng có độ dốc khá lớn.
* Nhẹ và mỏng, tỷ lệ sức mạnh / trọng lượng tốt hơn - tiết kiệm năng lượng.
* Khả năng lắp đặt dễ dàng và có độ bền cao.
* Với tính năng mềm dẻo, có độ đàn hồi tốt, có sức chịu đập mạnh, bề mặt nhẵn bóng đảm bảo băng tải PVC mang lại nhiều lợi ích cho doanh nghiệp khi sử dụng, dễ dàng di chuyển hàng từ nhiều vị trí khác nhau.

Nhược điểm:

* Vốn đầu tư ban đầu lớn.
* Một số hệ thống khó di chuyển, chỉ phù hợp với những lĩnh vực trong ngành công nghiệp nhẹ, không vận chuyển được các sản phẩm quá kích cỡ.
* Đối với các sản phẩm dạng hạt, vụn, … có thể bị hao hụt, rơi vã vật liệu trong quá trình vận chuyển.
* Khi vận chuyển xa và địa hình không thẳng đòi hỏi phải có nhiều hệ thống kết hợp lại với nhau.

### Chọn phương án thiết kế:

Tùy theo từng loại nguyên vật liệu hay sản phẩm mà có nhiều loại băng tải khác nhau: băng tải xích, băng tải con lăn, băng tải cao su băng tải pvc, băng tải đứng…

Trong trường hợp này chúng ta vận chuyển các chai thủy tinh, lon nhôm và chai nhựa nên chọn băng tải nhựa PVC (phương án 2) với những lí do sau:

* Hoàn toàn chống ẩm, chống gây mùi và nấm mốc.
* Dây băng tải hầu hết không thấm dầu mỡ và các hóa chất khác, đảm bảo độ an toàn khi chuyển tải hàng hóa, vật liệu.
* Độ bền kéo tốt, tuổi thọ dài, không bị dãn trong quá trình hoạt động.
* Hoạt động ổn định, năng suất vận hành cao, không gây tiêng ồn khi sử dụng.
* Tiết kiệm năng lượng.
* Khả năng lắp đặt dễ dàng và có độ bền cao.

Tuy nhiên băng tải này cũng có một số nhược điểm như là vốn đầu tư lớn, không vận chuyển được những vật thể quá kích thước và hệ thống khó di chuyển.

### Thiết kế băng tải:

Để tính toán và thiết kế hệ thống băng tải nhựa PVC, nhóm tác giả đã dựa vào tài liệu *Hướng dẫn tính toán băng tải – Nguyễn Văn Dự* *(2011).*

#### Khung băng tải:

* Thành khung băng tải:

Trên thị trường có 3 loại vật liệu phổ biến để chế tạo khung băng tải là thép sơn tĩnh điện, nhôm định hình, inox (thép không gỉ). Tùy theo nhu cầu sử dụng mà khách hàng có thể lựa chọn sao cho hợp lý.

Với những yêu cầu khung băng tải phải được thiết kế có sự chắc chắn, chịu lưc tốt. Phải chống oxi hóa, độ bền cao làm việc trong môi trường ẩm ước và có thể có hóa chất.

Từ những yêu cầu trên nhóm tác giả đã chọn vật liệu nhôm định hình.



Hình 3 - : Nhôm định hình

* Phần chân băng tải:

Chân của băng tải có tác dụng làm chân chống đỡ cho băng tải được hàn dính với thành băng tải để tạo cho khung băng tải có sự chắc chắn. Chân băng tải gồm các chân đứng, gân trợ lực sử dụng nhôm định hình.

#### Độ rộng tối thiểu băng tải:

Ở đây nhóm tác giả đang vận chuyển chai nhựa, chia thủy tinh và lon nhôm với thể tích nhỏ hơn 500ml và chiều dài nhỏ hơn 300 mm nên chọn băng tải có độ rộng 400 mm.

#### Góc nâng/ hạ của băng tải:

Góc nâng hay hạ của băng tải (góc dốc) được quyết định bởi đặc tính và hình dạng các hạt vật liệu được vận chuyển. Các vật liệu dạng hạt, ổn định có thể sử dụng băng tải có độ dốc lớn; các vật liệu không ổn định như than, cát cần xác lập góc dốc nhỏ.

Vật phẩm được vận chuyển là các lon nước, chai nhựa và chai thủy tinh nên gốc nâng hạ bằng 0.

#### Vận tốc băng tải:

Vận tốc băng tải cần giới hạn tùy thuộc dung lượng của băng, độ rộng của băng và đặc tính của vật liệu cần vận chuyển. Sử dụng băng hẹp chuyển động với vận tốc cao là kinh tế nhất; nhưng vận hành băng tải có độ rộng lớn lại dễ dàng hơn so với băng tải hẹp.

Vận tốc băng tải thường được tính toán nhằm đạt được lưu lượng vận chuyển theo yêu cầu cho trước. Lưu lượng vận chuyển của một băng tải có thể được xác định qua công thức:

Trong đó, Qt: Lưu lượng vận chuyển, tấn/ giờ;

* A: Diện tích mặt cắt ngang dòng vận chuyển (m2)
* γ: Khối lượng riêng tính toán của khối vật liệu (tấn/ m3)
* V: Vận tốc băng tải (m/phút)
* s: Hệ số ảnh hưởng của góc nghiêng (độ dốc) của băng tải

Từ đó, có thể tính được vận tốc băng tải theo công thức sau:

* Lưu lượng vân chuyển:
* Diện tích mặt cắt ngang dòng chảy:
  + Diện tích mặt cắt ngang dòng chảy có thể được xác định như sau:

Với A: Diện tích mặt cắt ngang dòng vận chuyển (m2)

* + - * B: Độ rộng băng tải (m)
      * K: Hệ số tính toán
  + Dựa vào bảng 4 (trang 7) thì hệ số tính toán sẽ được chọn:
    - Dạng băng tải: phẳng
    - Góc máng: 0
    - Góc mái: 10
  + K = 0,0295
* Diện tích mặt cắt ngang dòng chảy:

(m2)

* Khối lượng riêng tính toán:
  + Đối với các vật phẩm như chai nhựa, chai thủy tinh, lon nước khi được vận chuyển trên băng tải cách nhau 1 khoản cách nhất định thì có khối lượng riêng được tính riêng.
  + Với trung bình mỗi vật phẩm có khối lượng 0,1 kg, thể tích 500ml thì khối lượng riêng được tính là:
* Hệ số ảnh hưởng của độ dốc băng tải:
  + Băng tải càng dốc thì lưu lượng vận chuyển vật liệu được càng thấp.
  + Độ dốc của băng tải chúng ta lấy là 6 thì hệ số s = 0,98
* Vận tốc tối đa của băng tải là:

#### Tính toán công suất truyền dẫn băng tải:

Công suất làm quay trục con lăn kéo băng tải được tính theo công thức sau:

P = P1 + P2 + P3 + Pt (KW)

Trong đó:

* P1 là công suất cần thiết kéo băng tải không tải chuyển động theo phương ngang.
* P2 là công suất cần thiết kéo băng tải có chất tải chuyển động theo phương ngang.
* P3 là công suất kéo băng tải có tải chuyển động theo phương đứng (nếu băng tải có độ dốc đi lên; nếu băng tải vận chuyển vật phẩm đi xuống, P3 mang giá trị âm).
* Pt là công suất dẫn động cơ cấu gạt vật phẩm.

Băng tải chúng ta dùng là băng tải phẳng nên sẽ không có P3, và không cần cơ cấu gạt vật phẩm nên không có Pt.

Công suất P1 và P2 được tính theo công thức:

Trong đó:

* F: là hệ số ma sát của các ổ lăn đỡ con lăn
* W: là khối lượng các bộ phận chuyển động của băng tải, không tính khối lượng vật phẩm được vận chuyển (kg).
* Wm: Khối lượng vật phẩm phân bố trên một đơn vị dài của băng tải (kg/m).
* V: Vận tốc băng tải (m/phút)
* l: Chiều dài băng tải theo phương ngang (m)
* lo: Chiều dài băng tải theo phương ngang được điều chỉnh (m)

Các công thức tính phụ trợ:

Trong đó:

* Wl: Khối lượng phân bố của băng tải (kg/m)
* Wc: Khối lượng các chi tiết quay của một cụm các con lăn đỡ tải (kg);
* Wr: Khối lượng các chi tiết quay của một cụm các con lăn đỡ nhánh băng tải đi về;
* lc: Bước các con lăn đỡ tải (m)
* lr: Bước các con lăn đỡ nhánh chạy không (m)
* α: Góc dốc của băng tải
* Dựa vào bảng 8 (trang 11) với băng tải cố định thì:
  + F = 0,022
  + l0 = 66
* Dựa vào bảng 10 (trang 11) với chiều rộng đai 400(mm) thì:
  + Wl = 22 (kg/m)
* Dựa vào bảng 11 (trang 12) với chiều động đai 400(mm) thì:
  + Wc = 6,6 (kg/m)
  + Wr = 5,0 (kg/m)
* Dựa vào bảng 12 (trang 12) với chiều rộng đai 400(mm) và lưu lượng trên 1,6 tấn/giờ:
  + Bước con lăn mang băng tải lc = 1,35 (m)
  + Bước con lăn chạy không lr = 3 (m)
* Dựa vào bảng 13 (trang 13) với chiều rộng đai 400(mm) và khối lượng phân bố 0,3 kg/m thì:
  + Chiều dày gần đúng 9 (mm)
* Tính P1:
* Tính P2:
* Công suất truyền dẫn:

P = P1 + P2 = 0,74 + 0,008 = 0,748 (KW)

* Chọn P = 0,75 (KW)

#### Lực căng dây băng tải

* Lực vòng Fp:
* Lực căng trên 2 nhánh băng tải:

Trong đó:

* FP: lực vòng (kg).
* e: cơ số logarit tự nhiên.
* μ: hệ số ma sát giữa dây đai và pu-ly.
* θ: góc ôm giữa dây đai và pu-ly (radian).
* Dựa vào bảng 15 (trang 16) với dạng truyền dẫn đơn thì:
  + Gốc ôm θ = 180 (độ)
* Dựa vào bảng 16 (trang 16) với môi trường tiếp xúc puly là khô:
  + Hệ số ma sát μ = 0,3
* Lực căng trên 2 nhánh băng tải:
* Lực căng tối thiểu:
* Lực căng tối thiểu được xác định nhằm giữ cho dây băng tải không trượt quá 2% khoảng cách giữa các con lăn.
* Lực căng tối thiểu trên nhánh căng:
* Lực căng tối thiểu trên nhánh chùng:
* Lực kéo lớn nhất:
* Lực kéo lớn nhất được sử dụng để tính chọn dây băng tải theo độ bền. Đối với băng tải nằm ngang thì lực kéo lớn nhất được tính bằng:

Fmax = Fp + F2 = 76,5 + 48,8 = 125,3 (kg)

#### Tính chọn dây băng tải:

Trước hết, cần lưu ý về vật liệu dây băng tải. Dây băng tải cần được chọn phù hợp với môi trường làm việc, sao cho có thể tránh các ăn mòn hóa học.

Nhóm tác giả đang chọn đai dệt nhiều lớp, với công thức tính chọn:

Trong đó:

* Fmax là lực kéo lớn nhất (kg)
* SFz là hệ số an toàn
* Be là chiều rộng hữu ích của dây đai (cm)
* Dựa vào bảng 20 (trang 24) với:
  + Với chu kì 1-3 (chu kì / phút)
  + Kích thước vật thể lớn hơn 30mm
  + Vật liệu là các chai nhựa, chai thủy tinh và lon nhôm
* Hệ số an toàn là 11
* Dựa vào bảng 18, với F.TS = 34,36 và để đảm bảo an toàn ta chọn loại dải băng EP200.



Hình 3 - : Dải băng EP200

#### Cấu trúc hệ thống băng tải:

Các puly được chia thành 3 nhóm A, B và C:

* + Nhóm A: Puly dẫn động (Puly phía trước).
  + Nhóm B: Puly căng dây (Puly phía sau).
  + Nhóm C: Puly dẫn hướng.

Dựa vào bảng 23 (trang 25) với:

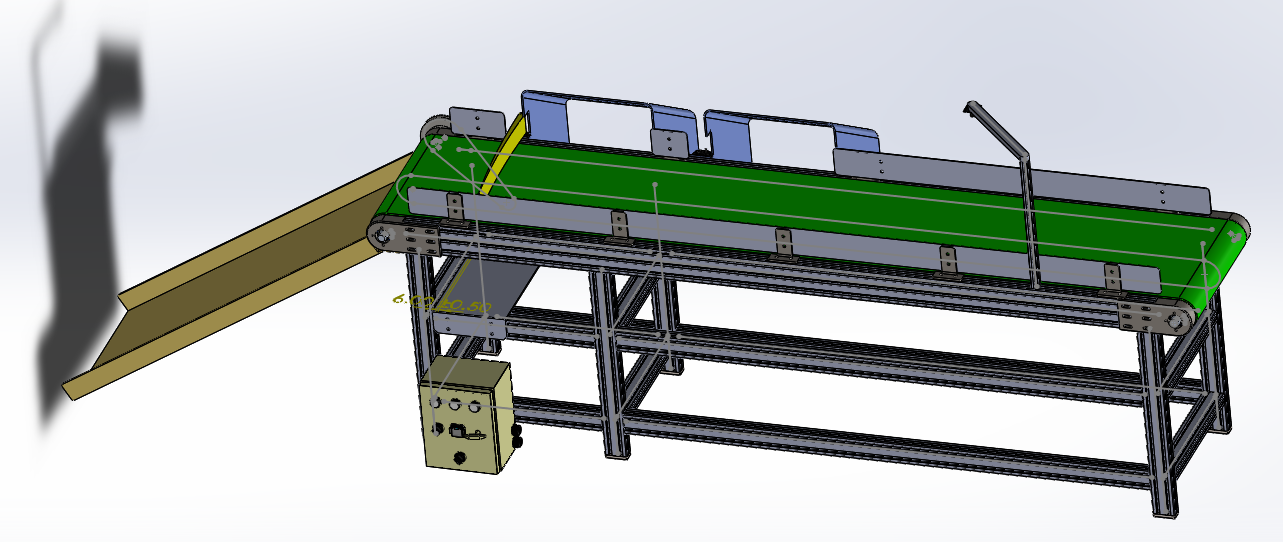
* Chịu kéo 30-60% khả năng cho phép.
* Với loại dải băng EP200.
* Đường kính loại A = 200, B = 160, C =125 (mm)

Nhưng vì chiều dài của băng tải chỉ 1m5 nên không cần đến Puly dẫn hướng, nên ta chọn đường kính Puly dẫn động và Puly căng dây bằng nhau và bằng 200 (mm)

Khoảng cách giữa các con lăn:

* Với độ rộng băng tải là 400 (mm)
* Khối lượng riêng của vật phẩm 200 kg/m­­­­­­­­3
* Khoảng cách là 1,7 lớn hơn chiều dài băng tải nên không có con lăn đỡ.

## Mô hình hệ thống:



Diagram, engineering drawing

Description automatically generated

Hình 3 - : Mô hình 3D hệ thống (Vẽ bằng phần mềm SolidWorks)

# chương 4: viết chương trình PLC, deep learning và giao diện, chức năng hệ thống SCADA

## Viết chương trình:

### Chương trình PLC:

Graphical user interface, text

Description automatically generated

Hình 4 - : Đoạn chương trình lập trình nút nhấn Start, Stop, Emergency

A picture containing graphical user interface

Description automatically generated

Hình 4 - : Đoạn chương trình lập trình chức năng watchdog và real time

Text

Description automatically generated with medium confidence

Hình 4 - : Đoạn chương trình lập trình hoạt động của băng tải

Diagram

Description automatically generated

Diagram, schematic

Description automatically generated

Hình 4 - : Đoạn chương trình lập trình chức năng cho hệ thống đèn chiếu sáng

A picture containing graphical user interface

Description automatically generated

Chart

Description automatically generated

Hình 4 - : Đoạn chương trình lập trình hoạt động của servo 1

Graphical user interface

Description automatically generated with medium confidence

Chart

Description automatically generated with medium confidence

Hình 4 - : Đoạn chương trình lập trình hoạt động của servo 2

Diagram

Description automatically generated

Hình 4 - : Đoạn chương trình lập trình cách phân loại sản phẩm thứ 3

A picture containing graphical user interface

Description automatically generated

Hình 4 - : Đoạn chương trình lập trình tín hiệu trigger đưa lên SCADA C# khi đã phân loại xong sản phẩm hiện tại

### Chương trình C#:

Vì chương trình C# rất dài nên ta sẽ tìm hiểu về đoạn mã chính phục vụ việc điều khiển và giám sát hệ thống.

using System;

using System.Windows.Forms;

using OPCAutomation;

namespace Mitsu\_SCADA\_WINFORM\_v6

{

public partial class Machine1 : Form

{

//int temperature = 0;

internal static Machine1 formMain;

public Machine1()

{

InitializeComponent();

formMain = this; // Make reference to formMain

}

private void Machine1\_Load(object sender, EventArgs e)

{

ConnectKEPServerEX();

}

//========================== Define objects of classes ==================

classStatusDisplay statusDisplay = new classStatusDisplay();

classLogin fnLogin = new classLogin();

//==========================KEPServerEX CONNECT=====================

static int tagNumber = 22; // Number of tags in this project

static int PLCscantime = 500; // Setting the time cycle to scan PLC

// Call connections to PLC

public OPCAutomation.OPCServer AnOPCServer;

public OPCAutomation.OPCServer OPCServer;

public OPCAutomation.OPCGroups OPCGroup;

public OPCAutomation.OPCGroup PLC;

public string Groupname;

static int arrlength = tagNumber + 1; // +1 because we not use arr[0]

Array OPtags = KEPServerEX.tagread(arrlength);

Array tagID = KEPServerEX.tagID(arrlength);

Array WriteItems = Array.CreateInstance(typeof(object), arrlength);

Array tagHandles = Array.CreateInstance(typeof(Int32), arrlength);

Array OPCError = Array.CreateInstance(typeof(Int32), arrlength);

Array dataType = Array.CreateInstance(typeof(Int16), arrlength);

Array AccessPaths = Array.CreateInstance(typeof(string), arrlength);

private void ConnectKEPServerEX()

{

string IOServer = "Kepware.KEPServerEX.V6";

string IOGroup = "OPCGroup1";

OPCServer = new OPCAutomation.OPCServer();

OPCServer.Connect(IOServer, "");

PLC = OPCServer.OPCGroups.Add(IOGroup);

PLC.DataChange += new DIOPCGroupEvent\_DataChangeEventHandler(DataScan);

PLC.UpdateRate = PLCscantime;

PLC.IsSubscribed = PLC.IsActive;

PLC.OPCItems.DefaultIsActive = true;

PLC.OPCItems.AddItems(tagNumber, ref OPtags, ref tagID,

out tagHandles, out OPCError, dataType, AccessPaths);

}

//========================== Read data from tags =====================

private void DataScan(int ID, int NumItems, ref Array tagID,

ref Array ItemValues, ref Array Qualities, ref Array TimeStamps)

{

for (int i = 1; i <= NumItems; i++)

{

// Identify tagID and tagValue being sent to App

int getTagID = Convert.ToInt32(tagID.GetValue(i));

string tagValue = ItemValues.GetValue(i)?.ToString();

// Get tag value

if (getTagID == 10)

{

statusDisplay.sttTwoStatus(symStart, tagValue);

}

if (getTagID == 11)

{

statusDisplay.sttTwoStatus(symStop, tagValue);

}

if (getTagID == 12)

{

statusDisplay.sttTwoStatus(symServo1, tagValue);

statusDisplay.sttTwoStatus(symContainer1, tagValue);

}

if (getTagID == 13)

{

statusDisplay.sttTwoStatus(symServo2, tagValue);

statusDisplay.sttTwoStatus(symContainer2, tagValue);

}

if (getTagID == 14)

{

statusDisplay.sttTwoStatus(symConveyor, tagValue);

}

if (getTagID == 15)

{

statusDisplay.sttTwoStatus(symLight1, tagValue);

statusDisplay.sttTwoStatus(symLight2, tagValue);

statusDisplay.sttTwoStatus(symLight3, tagValue);

}

if (getTagID == 16)

{

labTimeONLights.Text = tagValue;

}

if (getTagID == 17)

{

labTimeOFFLights.Text = tagValue;

}

if (getTagID == 18)

{

statusDisplay.sttTwoStatus(sensor1, tagValue);

}

if (getTagID == 19)

{

statusDisplay.sttTwoStatus(sensor2, tagValue);

}

}

}

//============== Transact data from popup window =============

public void popup\_button\_Clicked(int tagID)

{

WriteItems.SetValue(1, tagID);

PLC.SyncWrite(tagNumber, ref tagHandles, ref WriteItems, out OPCError);

WriteItems.SetValue(0, tagID);

}

private void symLight1\_Click(object sender, EventArgs e)

{

formPopupWindow frmPopup = new formPopupWindow();

frmPopup.tag\_ONID = 4; // ID tag Run

frmPopup.tag\_OFFID = 5; // ID tag Stop

frmPopup.Show();

formPopupWindow.formPopup.setTitle("Lights Control");

}

private void btnStart\_Click(object sender, EventArgs e)

{

WriteItems.SetValue(1, 1);

PLC.SyncWrite(tagNumber, ref tagHandles, ref WriteItems, out OPCError);

WriteItems.SetValue(0, 1);

}

private void btnStop\_Click(object sender, EventArgs e)

{

WriteItems.SetValue(1, 2);

PLC.SyncWrite(tagNumber, ref tagHandles, ref WriteItems, out OPCError);

WriteItems.SetValue(0, 2);

}

private void symServo1\_Click(object sender, EventArgs e)

{

formPopupWindow frmpopup = new formPopupWindow();

frmpopup.tag\_ONID = 6; // ID tag Run

frmpopup.tag\_OFFID = 7; // ID tag Stop

frmpopup.Show();

formPopupWindow.formPopup.setTitle("Servo 1 Control");

}

private void symServo2\_Click(object sender, EventArgs e)

{

formPopupWindow frmpopup = new formPopupWindow();

frmpopup.tag\_ONID = 8; // ID tag Run

frmpopup.tag\_OFFID = 9; // ID tag Stop

frmpopup.Show();

formPopupWindow.formPopup.setTitle("Servo 2 Control");

}

private void btnSaveLightTurnOnTime\_Click(object sender, EventArgs e)

{

WriteItems.SetValue(tbxLightTurnOnTime.Text, 16);

PLC.SyncWrite(tagNumber, ref tagHandles, ref WriteItems, out OPCError);

MessageBox.Show("Set Light turned ON time!");

}

private void btnSaveLightTurnOffTime\_Click(object sender, EventArgs e)

{

WriteItems.SetValue(tbxLightTurnOffTime.Text, 17);

PLC.SyncWrite(tagNumber, ref tagHandles, ref WriteItems, out OPCError);

MessageBox.Show("Set Light turned OFF time!");

}

}

}

### Chương trình python:

**import** argparse  
**import** os  
**import** platform  
**import** sys  
**from** pathlib **import** Path  
  
**import** pyodbc  
**import** torch  
  
FILE = Path(\_\_file\_\_).resolve()  
ROOT = FILE.parents[0] *# YOLOv5 root directory***if** str(ROOT) **not in** sys.path:  
 sys.path.append(str(ROOT)) *# add ROOT to PATH*ROOT = Path(os.path.relpath(ROOT, Path.cwd())) *# relative***from** models.common **import** DetectMultiBackend  
**from** utils.dataloaders **import** IMG\_FORMATS, VID\_FORMATS, LoadImages, LoadScreenshots, LoadStreams  
**from** utils.general **import** (Profile, check\_file, check\_img\_size, check\_imshow, check\_requirements, cv2,  
 increment\_path, non\_max\_suppression, print\_args, scale\_boxes, xyxy2xywh, LOGGER, colorstr,  
 strip\_optimizer)  
**from** utils.plots **import** Annotator, colors, save\_one\_box  
**from** utils.torch\_utils **import** select\_device, smart\_inference\_mode  
  
type\_product = 0  
path\_product = []  
count\_type = 0  
  
  
@smart\_inference\_mode()  
**def** run(  
 weights=ROOT / **'yolov5s.pt'**, *# model path or triton URL* source=ROOT / **'data/images'**, *# file/dir/URL/glob/screen/0(webcam)* data=ROOT / **'data/coco128.yaml'**, *# dataset.yaml path* imgsz=(640, 640), *# inference size (height, width)* conf\_thres=0.25, *# confidence threshold* iou\_thres=0.45, *# NMS IOU threshold* max\_det=1000, *# maximum detections per image* device=**''**, *# cuda device, i.e. 0 or 0,1,2,3 or cpu* view\_img=**False**, *# show results* save\_txt=**False**, *# save results to \*.txt* save\_conf=**False**, *# save confidences in --save-txt labels* save\_crop=**False**, *# save cropped prediction boxes* nosave=**False**, *# do not save images/videos* classes=**None**, *# filter by class: --class 0, or --class 0 2 3* agnostic\_nms=**False**, *# class-agnostic NMS* augment=**False**, *# augmented inference* visualize=**False**, *# visualize features* update=**False**, *# update all models* project=ROOT / **'runs/detect'**, *# save results to project/name* name=**'exp'**, *# save results to project/name* exist\_ok=**False**, *# existing project/name ok, do not increment* line\_thickness=3, *# bounding box thickness (pixels)* hide\_labels=**False**, *# hide labels* hide\_conf=**False**, *# hide confidences* half=**False**, *# use FP16 half-precision inference* dnn=**False**, *# use OpenCV DNN for ONNX inference* vid\_stride=1, *# video frame-rate stride*):  
 **global** type\_product, count\_type, path\_product  
  
 source = str(source)  
 save\_img = **not** nosave **and not** source.endswith(**'.txt'**) *# save inference images* is\_file = Path(source).suffix[1:] **in** (IMG\_FORMATS + VID\_FORMATS)  
 is\_url = source.lower().startswith((**'rtsp://'**, **'rtmp://'**, **'http://'**, **'https://'**))  
 webcam = source.isnumeric() **or** source.endswith(**'.txt'**) **or** (is\_url **and not** is\_file)  
 screenshot = source.lower().startswith(**'screen'**)  
 **if** is\_url **and** is\_file:  
 source = check\_file(source) *# download  
  
 # Directories* save\_dir = increment\_path(Path(project) / name, exist\_ok=exist\_ok) *# increment run* (save\_dir / **'labels' if** save\_txt **else** save\_dir).mkdir(parents=**True**, exist\_ok=**True**) *# make dir  
  
 # Load model* device = select\_device(device)  
 model = DetectMultiBackend(weights, device=device, dnn=dnn, data=data, fp16=half)  
 stride, names, pt = model.stride, model.names, model.pt  
 imgsz = check\_img\_size(imgsz, s=stride) *# check image size  
  
 # Dataloader* bs = 1 *# batch\_size* **if** webcam:  
 view\_img = check\_imshow(warn=**True**)  
 dataset = LoadStreams(source, img\_size=imgsz, stride=stride, auto=pt, vid\_stride=vid\_stride)  
 bs = len(dataset)  
 **elif** screenshot:  
 dataset = LoadScreenshots(source, img\_size=imgsz, stride=stride, auto=pt)  
 **else**:  
 dataset = LoadImages(source, img\_size=imgsz, stride=stride, auto=pt, vid\_stride=vid\_stride)  
 vid\_path, vid\_writer = [**None**] \* bs, [**None**] \* bs  
  
 *# Run inference* model.warmup(imgsz=(1 **if** pt **or** model.triton **else** bs, 3, \*imgsz)) *# warmup* seen, windows, dt = 0, [], (Profile(), Profile(), Profile())  
 **for** path, im, im0s, vid\_cap, s **in** dataset:  
 **with** dt[0]:  
 im = torch.from\_numpy(im).to(model.device)  
 im = im.half() **if** model.fp16 **else** im.float() *# uint8 to fp16/32* im /= 255 *# 0 - 255 to 0.0 - 1.0* **if** len(im.shape) == 3:  
 im = im[**None**] *# expand for batch dim  
  
 # Inference* **with** dt[1]:  
 visualize = increment\_path(save\_dir / Path(path).stem, mkdir=**True**) **if** visualize **else False** pred = model(im, augment=augment, visualize=visualize)  
  
 *# NMS* **with** dt[2]:  
 pred = non\_max\_suppression(pred, conf\_thres, iou\_thres, classes, agnostic\_nms, max\_det=max\_det)  
  
 *# Second-stage classifier (optional)  
 # pred = utils.general.apply\_classifier(pred, classifier\_model, im, im0s)  
 # print(pred)  
  
 # Process predictions* **for** i, det **in** enumerate(pred): *# per image* seen += 1  
 **if** webcam: *# batch\_size >= 1* p, im0, frame = path[i], im0s[i].copy(), dataset.count  
 s += **f'{**i**}: '  
 else**:  
 p, im0, frame = path, im0s.copy(), getattr(dataset, **'frame'**, 0)  
  
 p = Path(p) *# to Path* save\_path = str(save\_dir / p.name) *# im.jpg* txt\_path = str(save\_dir / **'labels'** / p.stem) + (**'' if** dataset.mode == **'image' else f'\_{**frame**}'**) *# im.txt* s += **'%gx%g '** % im.shape[2:] *# print string* gn = torch.tensor(im0.shape)[[1, 0, 1, 0]] *# normalization gain whwh* imc = im0.copy() **if** save\_crop **else** im0 *# for save\_crop* annotator = Annotator(im0, line\_width=line\_thickness, example=str(names))  
 **if** len(det):  
 *# Rescale boxes from img\_size to im0 size* det[:, :4] = scale\_boxes(im.shape[2:], det[:, :4], im0.shape).round()  
  
 *# Print results* **for** c **in** det[:, 5].unique():  
 n = (det[:, 5] == c).sum() *# detections per class* s += **f"{**n**} {**names[int(c)]**}{'s'** \* (n > 1)**}, "** *# add to string  
  
 # Write results* **for** \*xyxy, conf, cls **in** reversed(det):  
 **if** save\_txt: *# Write to file* xywh = (xyxy2xywh(torch.tensor(xyxy).view(1, 4)) / gn).view(-1).tolist() *# normalized xywh* line = (cls, \*xywh, conf) **if** save\_conf **else** (cls, \*xywh) *# label format* **with** open(**f'{**txt\_path**}.txt'**, **'a'**) **as** f:  
 f.write((**'%g '** \* len(line)).rstrip() % line + **'\n'**)  
  
 **if** save\_img **or** save\_crop **or** view\_img: *# Add bbox to image* c = int(cls) *# integer class* label = **None if** hide\_labels **else** (names[c] **if** hide\_conf **else f'{**names[c]**} {**conf**:.2f}'**)  
 annotator.box\_label(xyxy, label, color=colors(c, **True**))  
 **if** save\_crop:  
 save\_one\_box(xyxy, imc, file=save\_dir / **'crops'** / names[c] / **f'{**p.stem**}.jpg'**, BGR=**True**)  
  
 *# Stream results* im0 = annotator.result()  
 **if** view\_img:  
 **if** platform.system() == **'Linux' and** p **not in** windows:  
 windows.append(p)  
 cv2.namedWindow(str(p), cv2.WINDOW\_NORMAL | cv2.WINDOW\_KEEPRATIO) *# allow window resize (Linux)* cv2.resizeWindow(str(p), im0.shape[1], im0.shape[0])  
 cv2.imshow(str(p), im0)  
 cv2.waitKey(1) *# 1 millisecond  
  
 # Save results (image with detections)* **if** save\_img:  
 **if** dataset.mode == **'image'**:  
 cv2.imwrite(save\_path, im0)  
 **else**: *# 'video' or 'stream'* **if** vid\_path[i] != save\_path: *# new video* vid\_path[i] = save\_path  
 **if** isinstance(vid\_writer[i], cv2.VideoWriter):  
 vid\_writer[i].release() *# release previous video writer* **if** vid\_cap: *# video* fps = vid\_cap.get(cv2.CAP\_PROP\_FPS)  
 w = int(vid\_cap.get(cv2.CAP\_PROP\_FRAME\_WIDTH))  
 h = int(vid\_cap.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))  
 **else**: *# stream* fps, w, h = 30, im0.shape[1], im0.shape[0]  
 save\_path = str(Path(save\_path).with\_suffix(**'.mp4'**)) *# force \*.mp4 suffix on results videos* vid\_writer[i] = cv2.VideoWriter(save\_path, cv2.VideoWriter\_fourcc(\***'mp4v'**), fps, (w, h))  
 vid\_writer[i].write(im0)  
  
 *# Print time (inference-only)  
 # LOGGER.info(f"img{count\_type} : {s}{'' if len(det) else '(no detections), '}{dt[1].dt \* 1E3:.1f}ms")  
 # count\_type += 1* **if** len(det) == 1:  
 *# status = True* count\_type += 1  
 path\_product = det.numpy()  
 type\_product += path\_product[0][5]  
 *# LOGGER.info(f"img{count\_type} : {s} - {type\_product} - {dt[1].dt \* 1E3:.1f}ms")* **if** count\_type == 5:  
 *# print(f"----{type\_product/5}-----")* cursor = conx.cursor()  
 cursor.execute(  
 **"insert into Data(dayTime, typeProduct) values(getdate(),?)"**,  
 int(type\_product/5)  
 )  
 conx.commit()  
 **else**:  
 count\_type = 0  
 type\_product = 0 *# Print results* t = tuple(x.t / seen \* 1E3 **for** x **in** dt) *# speeds per image* LOGGER.info(**f'Speed: %.1fms pre-process, %.1fms inference, %.1fms NMS per image at shape {**(1, 3, \*imgsz)**}'** % t)  
 **if** save\_txt **or** save\_img:  
 s = **f"\n{**len(list(save\_dir.glob(**'labels/\*.txt'**)))**} labels saved to {**save\_dir / **'labels'}" if** save\_txt **else ''** LOGGER.info(**f"Results saved to {**colorstr(**'bold'**, save\_dir)**}{**s**}"**)  
 **if** update:  
 strip\_optimizer(weights[0]) *# update model (to fix SourceChangeWarning)***def** parse\_opt():  
 parser = argparse.ArgumentParser()  
 parser.add\_argument(**'--weights'**, nargs=**'+'**, type=str, default=ROOT / **'yolov5s.pt'**, help=**'model path or triton URL'**)  
 parser.add\_argument(**'--source'**, type=str, default=ROOT / **'data/images'**, help=**'file/dir/URL/glob/screen/0(webcam)'**)  
 parser.add\_argument(**'--data'**, type=str, default=ROOT / **'data/coco128.yaml'**, help=**'(optional) dataset.yaml path'**)  
 parser.add\_argument(**'--imgsz'**, **'--img'**, **'--img-size'**, nargs=**'+'**, type=int, default=[640], help=**'inference size h,w'**)  
 parser.add\_argument(**'--conf-thres'**, type=float, default=0.25, help=**'confidence threshold'**)  
 parser.add\_argument(**'--iou-thres'**, type=float, default=0.45, help=**'NMS IoU threshold'**)  
 parser.add\_argument(**'--max-det'**, type=int, default=1000, help=**'maximum detections per image'**)  
 parser.add\_argument(**'--device'**, default=**''**, help=**'cuda device, i.e. 0 or 0,1,2,3 or cpu'**)  
 parser.add\_argument(**'--view-img'**, action=**'store\_true'**, help=**'show results'**)  
 parser.add\_argument(**'--save-txt'**, action=**'store\_true'**, help=**'save results to \*.txt'**)  
 parser.add\_argument(**'--save-conf'**, action=**'store\_true'**, help=**'save confidences in --save-txt labels'**)  
 parser.add\_argument(**'--save-crop'**, action=**'store\_true'**, help=**'save cropped prediction boxes'**)  
 parser.add\_argument(**'--nosave'**, action=**'store\_true'**, help=**'do not save images/videos'**)  
 parser.add\_argument(**'--classes'**, nargs=**'+'**, type=int, help=**'filter by class: --classes 0, or --classes 0 2 3'**)  
 parser.add\_argument(**'--agnostic-nms'**, action=**'store\_true'**, help=**'class-agnostic NMS'**)  
 parser.add\_argument(**'--augment'**, action=**'store\_true'**, help=**'augmented inference'**)  
 parser.add\_argument(**'--visualize'**, action=**'store\_true'**, help=**'visualize features'**)  
 parser.add\_argument(**'--update'**, action=**'store\_true'**, help=**'update all models'**)  
 parser.add\_argument(**'--project'**, default=ROOT / **'runs/detect'**, help=**'save results to project/name'**)  
 parser.add\_argument(**'--name'**, default=**'exp'**, help=**'save results to project/name'**)  
 parser.add\_argument(**'--exist-ok'**, action=**'store\_true'**, help=**'existing project/name ok, do not increment'**)  
 parser.add\_argument(**'--line-thickness'**, default=3, type=int, help=**'bounding box thickness (pixels)'**)  
 parser.add\_argument(**'--hide-labels'**, default=**False**, action=**'store\_true'**, help=**'hide labels'**)  
 parser.add\_argument(**'--hide-conf'**, default=**False**, action=**'store\_true'**, help=**'hide confidences'**)  
 parser.add\_argument(**'--half'**, action=**'store\_true'**, help=**'use FP16 half-precision inference'**)  
 parser.add\_argument(**'--dnn'**, action=**'store\_true'**, help=**'use OpenCV DNN for ONNX inference'**)  
 parser.add\_argument(**'--vid-stride'**, type=int, default=1, help=**'video frame-rate stride'**)  
 opt = parser.parse\_args()  
 opt.imgsz \*= 2 **if** len(opt.imgsz) == 1 **else** 1 *# expand* print\_args(vars(opt))  
 **return** opt  
  
  
**def** main(opt):  
 check\_requirements(exclude=(**'tensorboard'**, **'thop'**))  
 run(\*\*vars(opt))  
  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 conx = pyodbc.connect(  
 **'DRIVER={ODBC Driver 13 for SQL Server}; SERVER=DESKTOP-7C40RE7\SQLEXPRESS; Database=DoAnTotNghiep; UID=sa; PWD=123456'**)  
  
 opt = parse\_opt()  
 main(opt)

## Chức năng hệ thống SCADA:

### Quản lý người dùng:

### Giám sát:

### Điều khiển:

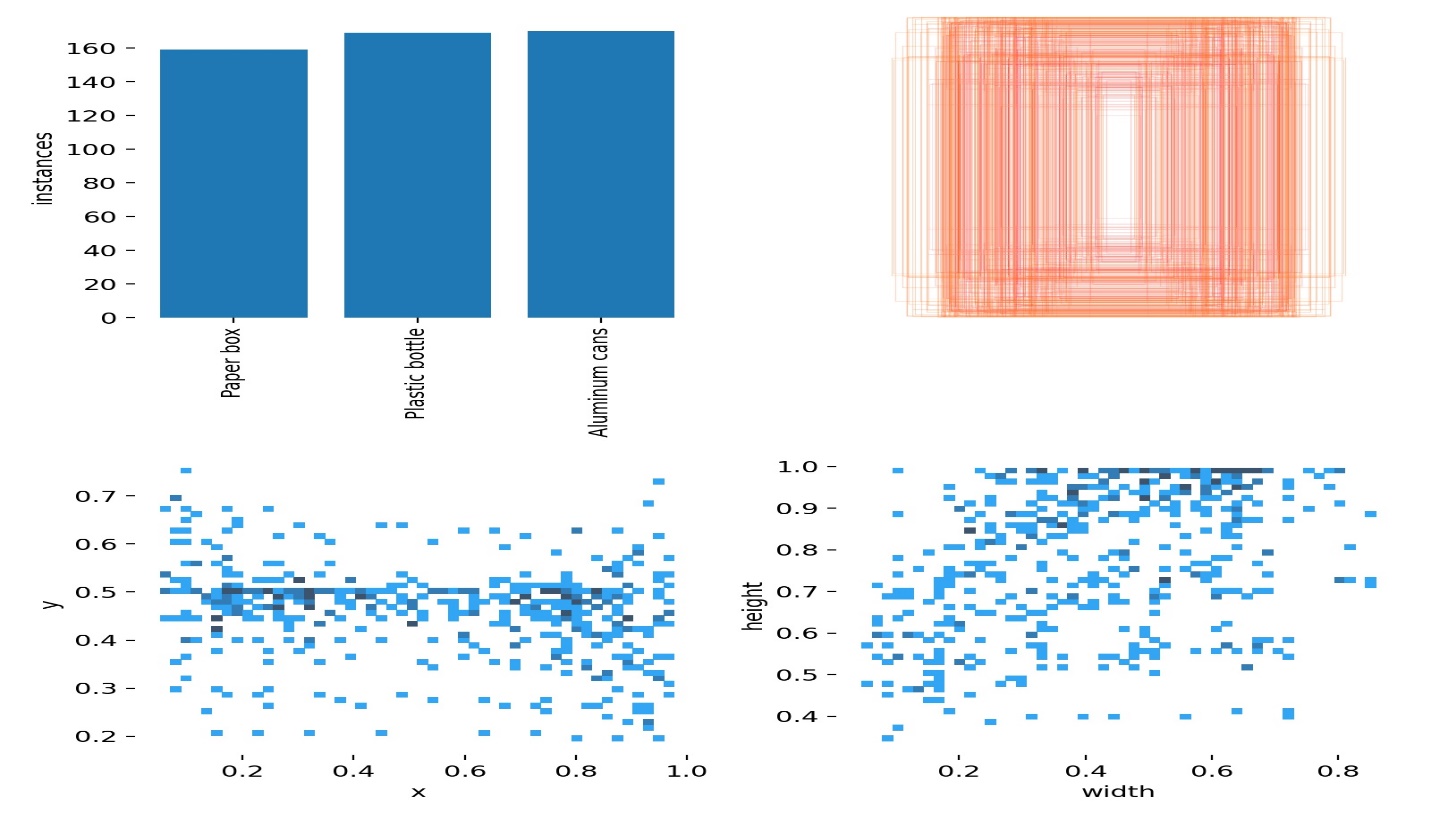
### Trạng thái cảm biến:

### Watchdog:

### Quản trị cơ sở dữ liệu:

### Xuất báo cáo file PDF, EXCEL, WORD:

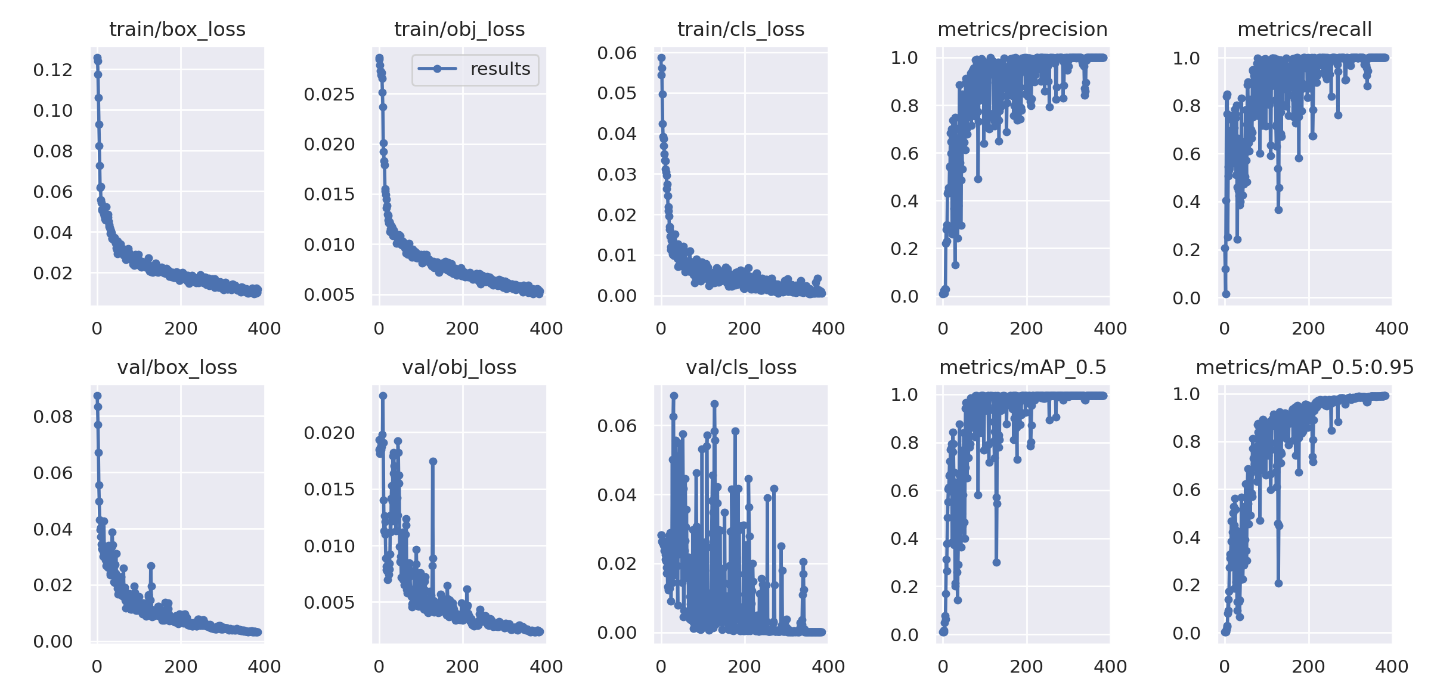
## Phân loại bằng công nghệ xử lý ảnh:



Hình 4 - : Phân loại 3 loại rác tái chế bằng công nghệ xử lý ảnh

Nhóm tác giả sử dụng module YOLOv5 để training data và phân loại rác tái chế dựa vào hình ảnh nhận được từ camera.

Module YOLOv5 viết bằng ngôn ngữ Python và chạy bằng phần mềm PyCharm 2021.3.3



Hình 4 - : Biểu đồ sai số (loss) và độ chính xác (precision) khi training data

Tổng số dữ liệu được training lên đến 500 hình ảnh, với gần 200 hình mỗi loại rác, mang đến độ chính xác gần như tuyệt đối khi phân loại một trong 3 loại: hộp giấy, hộp nhựa và lon nước.

# chương 5: kết luận

* Việc ứng dụng công nghệ học sâu vào hệ thống phân loại giúp cải thiện hiệu suất của máy, giảm nguồn lực con người làm những việc nhàm chán.
* Ứng dụng SCADA viết bằng ngôn ngữ C# điều khiển, giám sát một cách nhanh, gọn, nhẹ nhờ KepwareOPC.
* Nhóm tác giả mong muốn áp dụng những công nghệ mới ở thời đại 4.0 vào việc cải thiện môi trường sống con người.
* Vì thời gian nghiên cứu và tình hình kinh tế còn hạn hẹp, chưa thể làm máy hoàn chỉnh, nhưng thiết nghĩ mô hình cũng đủ để các độc giả hình dung được hệ thống.

## Giới hạn đề tài:

* Hệ thống chỉ cho phép điều khiển và giám sát cục bộ, chưa thể áp dụng công nghệ WebServer để điều khiển và giám sát từ xa vì thời gian và kinh phí có hạn.
* Tốc độ phân loại chưa đạt mục tiêu đề ra vì các cơ cấu và cảm biến chỉ dừng lại ở mức độ mô phỏng.

## Hướng phát triển:

* Có thể dễ dàng mở rộng hệ thống bằng việc tăng số lượng băng tải, hoặc tăng số loại rác tái chế phân loại được trên mỗi băng tải.
* Có thể áp dụng những công nghệ trong hệ thống để phân loại sản phẩm, hàng háo trong sản xuất đạt năng suất cao.
* Có thể dễ dàng nâng cấp phiên bản SCADA từ mạng cục bộ sang internet bằng SCADA webserver. Từ đó có thể giám sát, điều khiển ở bất cứ đâu có internet.

# tài liệu tham khảo

[1]: Manual của PLC Mitsubishi FX1N - [*http://dl.mitsubishielectric.com/dl/fa/document/manual/plc\_fx/jy992d88101/jy992d88101e.pdf*](http://dl.mitsubishielectric.com/dl/fa/document/manual/plc_fx/jy992d88101/jy992d88101e.pdf)

[2] Báo Nhân Dân - [*https://nhandan.vn/thuc-day-tai-che-rac-de-giai-quyet-van-de-o-nhiem-do-rac-thai-post623467.html*](https://nhandan.vn/thuc-day-tai-che-rac-de-giai-quyet-van-de-o-nhiem-do-rac-thai-post623467.html)

[3] Trang thông tin điện tử xã hội -[*https://soha.vn/gam-mau-sang-toi-ve-thuc-trang-rac-thai-nhua-viet-nam-giai-phap-tat-yeu-trung-hoa-nhua-20211207212608163.htm*](https://soha.vn/gam-mau-sang-toi-ve-thuc-trang-rac-thai-nhua-viet-nam-giai-phap-tat-yeu-trung-hoa-nhua-20211207212608163.htm)

[4] Báo Thanh Niên - [*https://thanhnien.vn/thuc-trang-rac-thai-nhua-viet-nam-giai-phap-tat-yeu-trung-hoa-nhua-post1410396.html*](https://thanhnien.vn/thuc-trang-rac-thai-nhua-viet-nam-giai-phap-tat-yeu-trung-hoa-nhua-post1410396.html)

[5] Cách cài phần mềm và giải pháp thiết kế SCADA cho lĩnh vực tự động hóa tham khảo website: [*https://ngocautomation.com/*](https://ngocautomation.com/)

[6] Hướng dẫn tính toán băng tải *– Nguyễn Văn Dự* *(2011).*