



POP 스마트 생산시점관리 시스템

이승민

유현수(발표자)

2025.07.28

목차

프로젝트 개요

기획 및 설계

주요 기능

개발 환경 도구

브랜치 전략

시연 및 트러블 슈팅

개선점

프로젝트 개요

PCB 생산 현장의 설비 상태 및 생산 실적을 실시간으로 모니터링하고, 불량 데이터를 수집하여 공정 효율을 향상시키기 위해 POP 시스템을 개발하였습니다

사용자 중심 인터페이스

C# WPF 기술을 활용한 직관적인 모니터링 및 제어 인터페이스 개발

자동화 시스템 개발

PLC 기반 생산 공정 자동화 시스템을 구축하여 제조 효율성 향상 및 인적 오류 최소화

품질 검사 자동화

OpenCV 기반 영상 처리 기술로 실시간 제품 품질 검사 및 불량품 자동 식별 시스템 구현

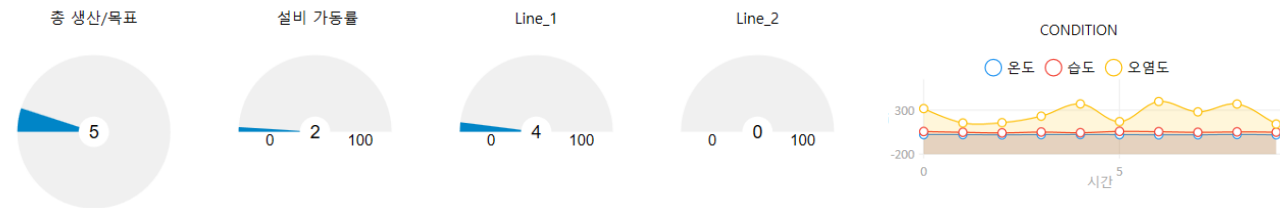
데이터 기반 생산 활동 기록

통합 데이터베이스 시스템을 통한 생산 데이터 수집, 분석

주요 기능

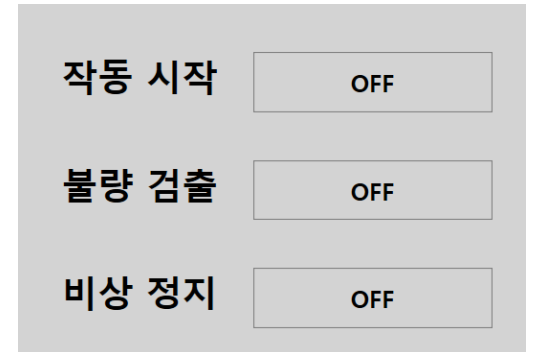
1

C1.WPF.Chart, LiveChart 라이브러리를 활용하여 생산 현황 및 공장 환경 데이터를 효과적으로 전달



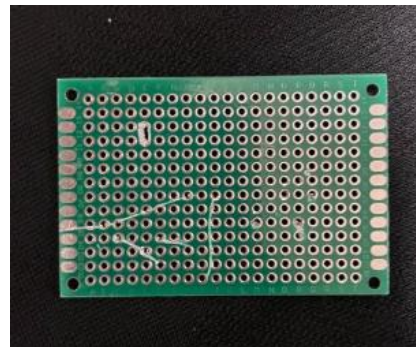
2

PLC 래더 다이어그램을 Modbus 통신으로 WPF에서 직접 제어



3

OPENCV를 활용하여 QR CODE 정보, 반/완제품별 불량을 분석하고 데이터를 저장



4

C#과 DB의 동기화로 불량 검출 시 램프 점멸

```
// 1. 제품의 불량 여부 확인 (Defective_or_not == true)
if (CurrentProduct?.Defective_or_not == true)
{
    try
    {
        // 2. PLC 코일 5번을 ON
        await modbus.WriteCoilAsync(5, true);

        // 3. 6초 대기
        await Task.Delay(6000);

        // 4. 6초 후 PLC 코일 5번을 OFF
        await modbus.WriteCoilAsync(5, false);
    }
    catch (Exception ex)
    {
        MessageBox.Show($"PLC 제어 실패 (불량 검출): {ex.Message}", "Modbus 통신 오류", MessageBoxButton.OK, MessageBoxImage.Error);
    }
}
else
{
    // 정상 제품인 경우
    Debug.WriteLine($"정상 제품입니다 ({CurrentProduct.QR_Code}). 병드 조치 없음.");
}
```

기획 및 설계



UI 설계

C# WPF 기술을 활용한 직관적인 모니터링 및 MVVM 패턴으로 유지보수성 향상



PLC 통신

PLC 기반 생산 공정 자동화 시스템을 구축하여 제조 효율성 향상 및 인적 오류 최소화



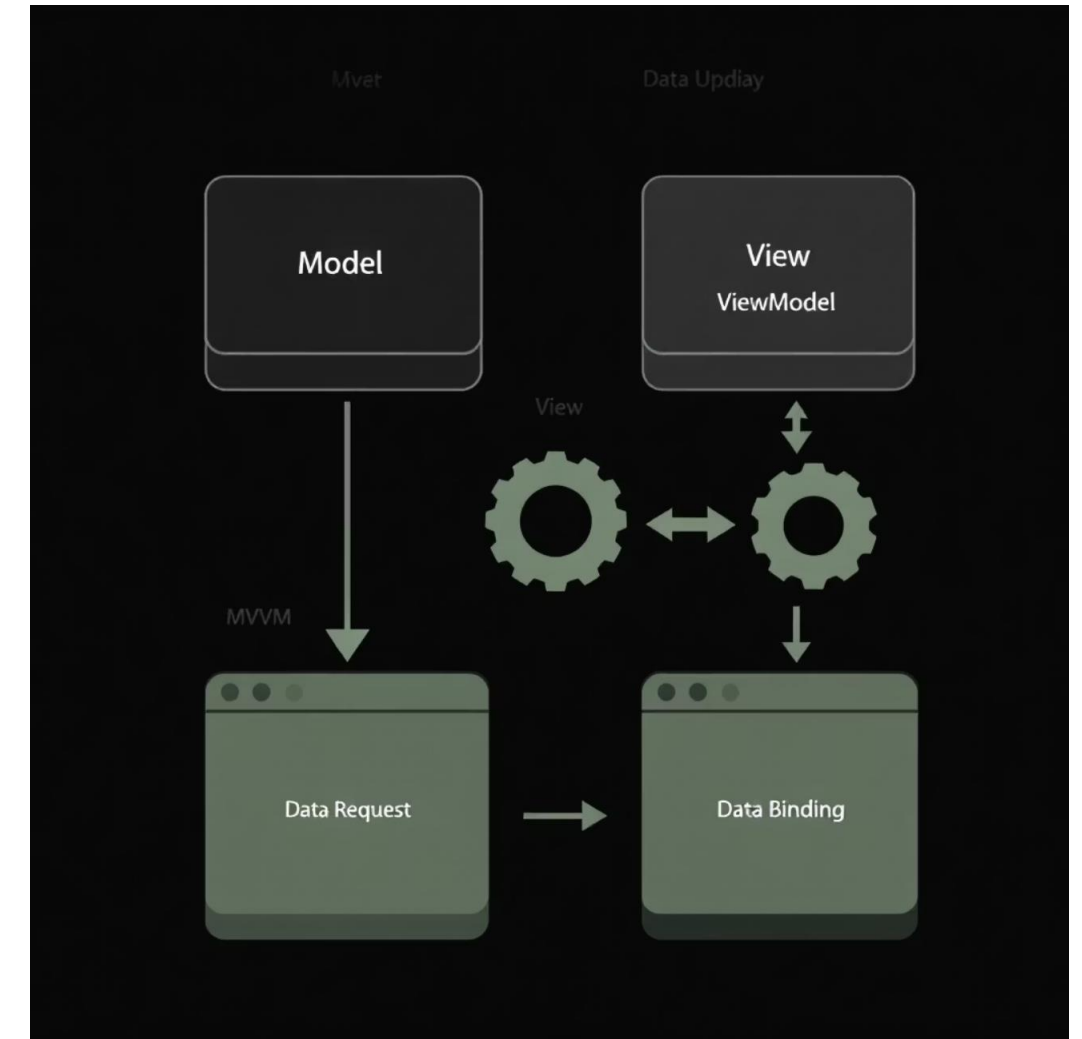
이미지 처리

OpenCV 기반 영상 처리 기술로 실시간 제품 품질 검사 및 불량품 식별 시스템



데이터 관리

MySQL DB 설계: 생산, 불량, 설비, 환경, 유저 테이블 관계 정의



기획 및 설계



product		
qr_code	qr_code	int
name	name	Varchar(30)
status	status	Varchar(20)
production_time	production_time	Datetime
defective_or_not	defective_or_not	bool

defect		
qr_code	qr_code	int
status	status	Varchar(20)
time_defect	time_defect	Datetime
location	location	Tinyint
name	name	Varchar(20)
class_defect	class_defect	Varchar(20)

users		
id	id	Varchar(30)
pwd	pwd	Varchar(30)
name	name	Varchar(30)
handover	handover	TEXT

facility		
name	name	Varchar(20)
running_time	running_time	int
recent_check	recent_check	Datetime
manufacturer	manufacturer	Varchar(20)
Model_name	Model_name	Varchar(20)
Field	Domain	Type

Condition		
temperature	temperature	double
humidity	humidity	int
pollution	pollution	int
powerStability	powerStability	tinyint(1)
create_date	create_date	timestamp

⚠ 전체 이슈 상황

담당자	To do list	Aa Troubleshooting	시작	완료	비고	구현 상태
유현수	● PLC 래더 다이어그램 작성	📄 램프 플리커 문제 ⇒ COUNT_INT 평선 블록을 사용하여 3회 점멸 후 on 유지	2025/07/02	2025/07/03	램프(출력): 벨트 가동 중지 시 점멸	✓
이승민	● OpenCV 스마트폰 연동/...	📄	2025/07/03	2025/07/03		✓
이승민	● OpenCV QR리더기 함수...	📄	2025/07/02	2025/07/06		✓
유현수	● FIGMA 그림/ 기획서 작성		2025/07/03	2025/07/04		✓
이승민	● QR데이터 로직		2025/07/04	2025/07/15		✓
이승민	● python-MySQL 연동		2025/07/06	2025/07/06		✓
이승민	● 불량 검출 로직	📄	2025/07/09	2025/07/15		✓
유현수	● C# WPF UI		2025/07/04	2025/07/08		✓
유현수	● MYSQL-C# 연동		2025/07/08	2025/07/11	ERDCloud 사용해서 쿼리문 작성	✓
이승민	● yolo모델 학습		2025/07/08	2025/07/13		✓
유현수	● PCB 기판 완제품 소자 배...		2025/07/09	2025/07/09		✓
유현수	● PLC 통신 연결	📄 PLC IP문제 ⇒ Ethernet 통신을 위한 PLC의 IP 주소 게이트가 1로 되어있어서 0으로 변경 후 쓰기	2025/07/14	2025/07/15		✓
이승민	● C# 데이터 정리 및 도식화	📄	2025/07/19	2025/07/25		✓
유현수	● Modbus로직	📄 C#과 연결 과정에서 변수 매핑오류 ⇒ 기존 모드버스 설정의 비트 읽기, 쓰기 영역을 %MX0으로 수정하고 ModbusControl의 ip, port를 정확히 명시	2025/07/21	2025/07/25		✓
유현수	● 공장 환경 LiveChart		2025/07/25	2025/07/25		✓

개발 환경 및 도구

프로그래밍 환경

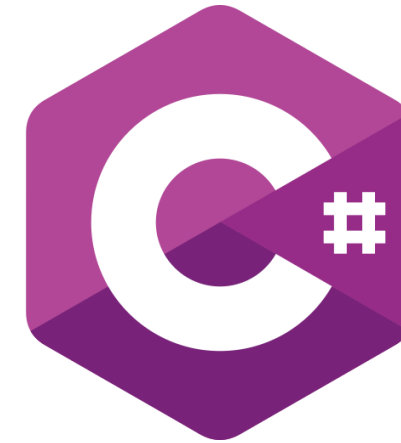
- C# (.NET Framework 4.8)
- Python 3.13.5
- WPF MVVM 프레임워크

하드웨어 제어

- LS XGI CPUE PLC
- Modbus TCP/IP 프로토콜
- 휴대폰 카메라

영상처리 및 데이터

- OpenCV 4.11.0.86
- MySQL 서버



브랜치 전략 및 협업 프로세스

브랜치명	작업 영역	역할	전략
main		최종 결과물을 담는 브랜치(realease)	안정적인 코드만 유지하도록 주의
main_develop		각 브랜치 병합 후 main 이동 전 테스트 브랜치	각 브랜치에서 통합 완료 한 뒤 병합
OPENCV		불량 판별 처리 및 신호 전달	영상 인식 YOLO 학습
	OPENCV_sm		각자 작업한 내용 푸쉬하기
	OPENCV_hs		각자 작업한 내용 푸쉬하기
PLC		통신, 컨베이어 벨트 제어	벨트, 램프 제어 로직 기능
	PLC_conveyer		컨베이어의 가동, 정지 ... 속도 기능
	PLC_lamp		제품 불량 판별로 램프 작동 방식 로직
XAML		전체 진행 상태와 DB로 불량 상태 시각화	MVVM 구조, Repository 패턴 기반 ViewModel, DB 연동
	XAML_sm		각자 작업한 내용 푸쉬하기
	XAML_hs		각자 작업한 내용 푸쉬하기

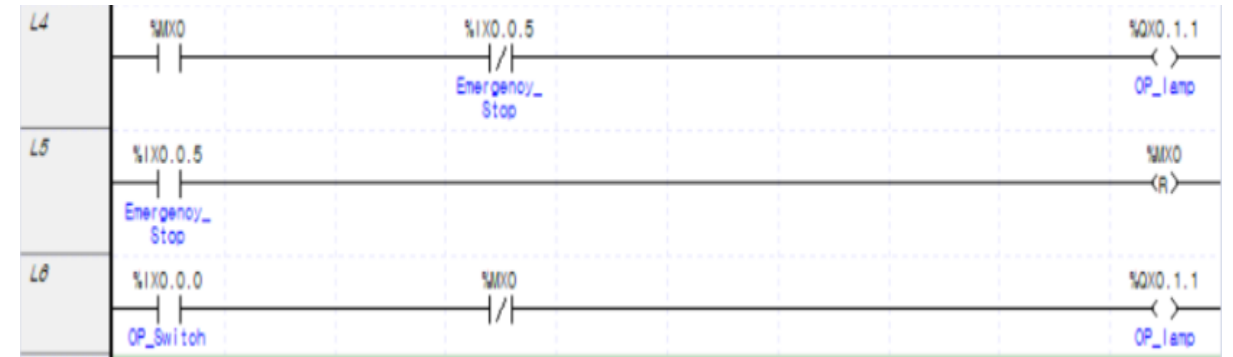
트러블 슈팅

LD 이중코일

램프 ON/OFF 상태
이중코일로 인한 오류

1

=> 램프1, 2로 불량 검출 램프
프를 분리하여 직관성 향상

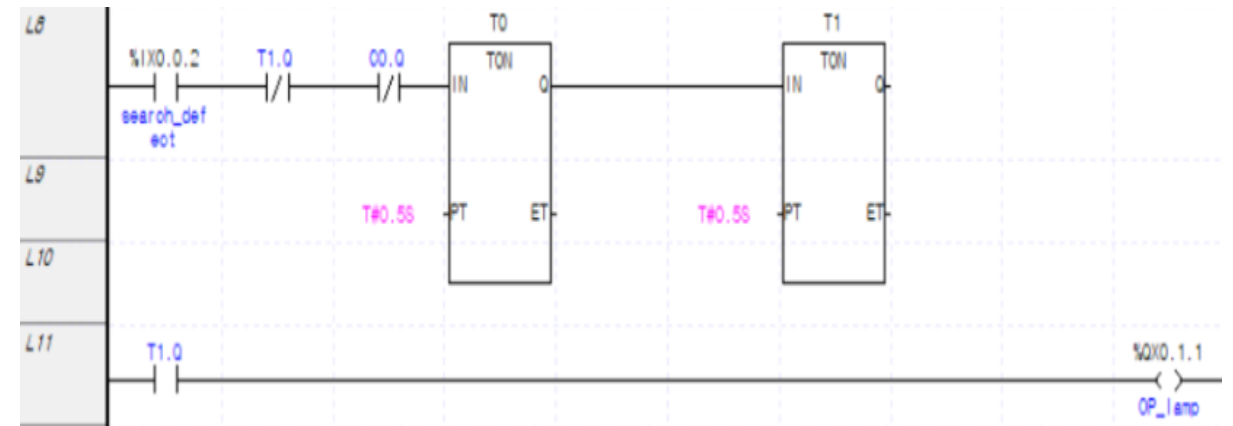


PLC IP 대역

Ethernet 통신 과정에서 PLC와 공
공유기, PC의 IP 대역이 다름

2

=> PLC IP 대역을 0으로 변경
하여 통일



Modbus 로직

C#과 연동하여 제어하는
과정에서 변수 매핑 오류

3

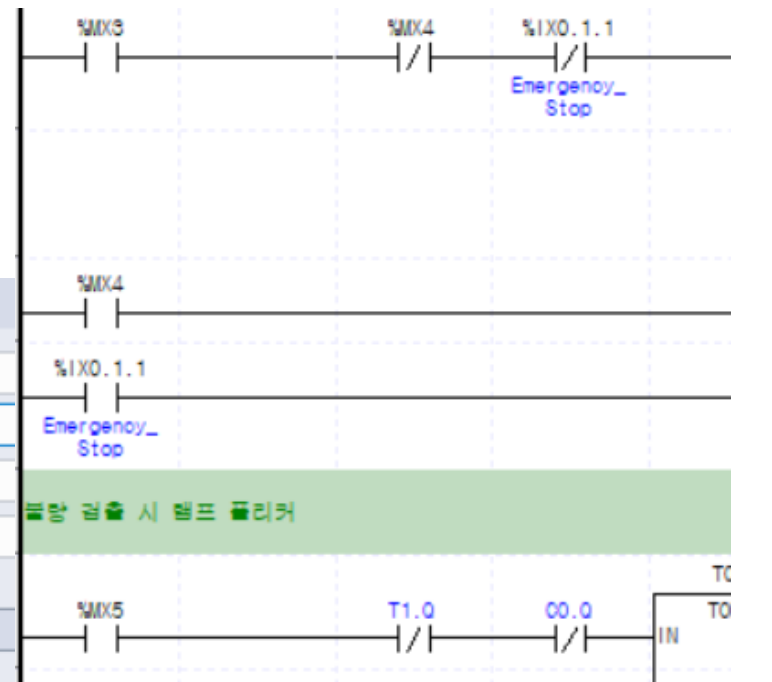
=> Modbus 비트 읽기, 쓰기 영역
역을 메모리 타입인 %MX0~로 보
보완하고 ModbusControl ip, port
port 명시

Modbus 설정

비트 읽기 영역 시작 주소: %MX0
비트 쓰기 영역 시작 주소: %MX0
워드 읽기 영역 시작 주소: %MW0
워드 쓰기 영역 시작 주소: %MW100

확인

취소



트러블 슈팅

- 순환 참조

순환 참조가 일어나 서로 의존하는 관계가 만들어지며 Error가 생김

원인 : 서로 import할 때 파이썬이 아직 정의되지 않은 객체에 접근하려 해서 발생

해결 : OpenCV를 키는 파일에 QR코드 리더기 로직을 추가하여 기능 통합

```
문제  출력 디버그 콘솔 터미널 포트 GITLENS

/OpenCV_SmartPhone.py
Traceback (most recent call last):
  File "c:\Users\o\Desktop\POP\POP\OPENCV_Python\OpenCV_SmartPhone.py", line 3, in <module>
    from OpenCV_QRreader import decode_qr_code
  File "c:\Users\o\Desktop\POP\POP\OPENCV_Python\OpenCV_QRreader.py", line 10, in <module>
    from OpenCV_SmartPhone import run_camera
  File "c:\Users\o\Desktop\POP\POP\OPENCV_Python\OpenCV_SmartPhone.py", line 3, in <module>
    from OpenCV_QRreader import decode_qr_code
ImportError: cannot import name 'decode_qr_code' from 'OpenCV_QRreader' (consider renaming 'c:\\Users\\o\\Desktop\\POP\\POP\\OPENCV_Python\\OpenCV_QRreader.py' if it has the same name as a library you intended to import)
```

트러블 슈팅

- QR코드 read 반복

1차 이후 location이 1로 바뀌면서 2차가 즉시 진행되는 오류

원인 : 1차 불량 검사와 2차 불량 검사에 대한 딜레이가 없음

결과 : qr에 반제품/완제품을 미리 구별하여 DB에 UPDATE 후 반제품일때 → 1차 검사만 진행
완제품일때 → 2차 검사만 진행

```
인식 성공 : 41  
1차 검사 시작  
  
0: 480x640 1 scratch, 201.3ms  
Speed: 7.7ms preprocess, 201.3ms inference, 2.1ms postprocess per image at shape (1, 3, 480, 640)  
Detected scratch with confidence 0.34 at [403, 332, 421, 361]  
인식 성공 : 41  
2차 검사 시작  
  
0: 480x640 (no detections), 193.8ms  
Speed: 8.4ms preprocess, 193.8ms inference, 1.3ms postprocess per image at shape (1, 3, 480, 640)
```

트러블 슈팅

- Git 업로드 및 충돌 문제

.vs, bin, obj 등의 불필요한 폴더가 Github에 올라감

원인 : .gitignore 설정 미흡 및 캐시 파일을 제외하지 않음

해결 : .gitignore 파일 작성 후 git rm -cached로 Git에 스테이징 된
부분 삭제 및 .gitignore 파일 수정

bin/ obj/	*.pdb *.cache *.dbmdl *.vsp *.vspssc *.vssssc
*.g.cs *.g.i.cs *.designer.cs	project.lock.json project.assets.json *.nupkg packages/ .nuget/
*.user *.suo *.userosscache *.sln.docstates *.vs/	.idea/ .vscode/
*.tmp *.log *.bak	Thumbs.db ehthumbs.db .DS_Store *.swp
*.pdb *.cache *.dbmdl *.vsp *.vspssc *.vssssc	_ReSharper*/ *.resharper *.DotSettings.user *.dll
project.lock.json project.assets.json *.nupkg packages/ .nuget/	

프로젝트 개선점 및 향후 계획



1단계: 추정치 기반 더미데이터

실측 데이터 확보를 통해 품질 검증



2단계: 기능 확장

컨베이어 속도, 방향 등 추가 제어 기능 구현



3단계: 신뢰도 최적화

불량 검출 정확도 향상을 위한 추가 학습