

# Introduction

로보틱스설계를 통한 스마트제조 가상공장구현

# 아두이노를 통한 IoT 센서 활용

## 주요 사용 부품

- 포토센서 (예: LDR 또는 IR 센서)
- 서보모터 (예: SG90)
- LED (상태 표시용) 온습도 센서 (DHT22, BME280)
- 진동 센서 (ADXL345)
- 전류 센서 (ACS712)
- 압력 센서 (BMP280)

# 아두이노를 통한 IoT 센서 활용

```
1  #include <Servo.h>
2
3  #define PHOTO_SENSOR_PIN A0
4  #define SERVO_PIN 9
5  #define LED_PIN 13
6
7  Servo sortingServo;
8  int sensorThreshold = 500; // 조정 필요한 threshold 값
9
10 void setup() {
11     Serial.begin(9600);
12     pinMode(LED_PIN, OUTPUT);
13     sortingServo.attach(SERVO_PIN);
14     sortingServo.write(90); // 초기 위치
15 }
16
17 void loop() {
18     int sensorValue = analogRead(PHOTO_SENSOR_PIN);
19     Serial.print("Sensor value: ");
20     Serial.println(sensorValue);
21
22     if (sensorValue < sensorThreshold) {
23         // 제품 감지
24         digitalWrite(LED_PIN, HIGH);
25         sortProduct();
26     } else {
27         digitalWrite(LED_PIN, LOW);
28     }
```

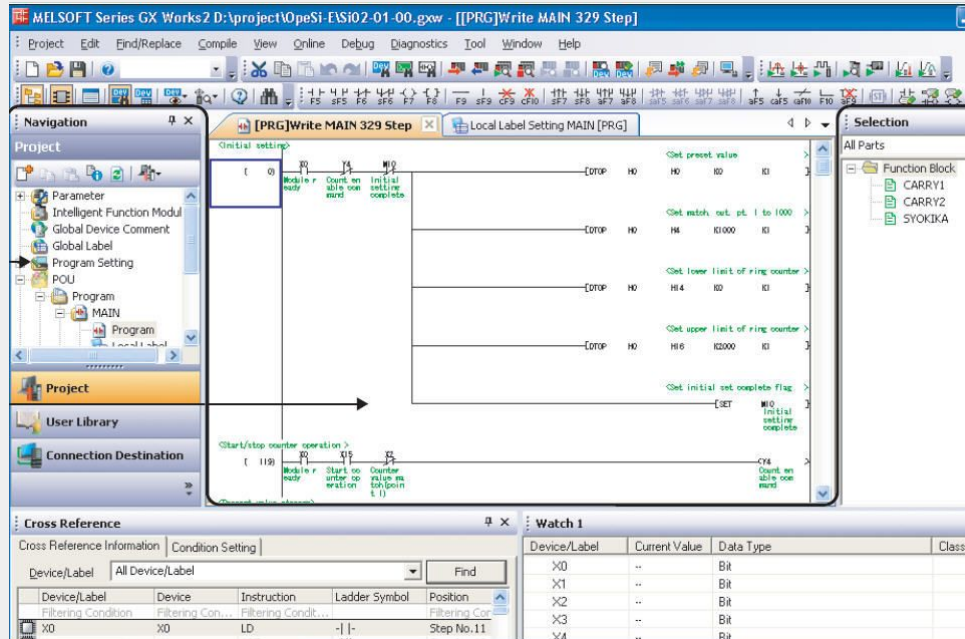
```
1  #include <DHT.h>
2  #define DHTPIN 2
3  #define DHTTYPE DHT22
4  DHT dht(DHTPIN, DHTTYPE);
5
6  void setup() {
7     Serial.begin(9600);
8     dht.begin();
9 }
10
11 void loop() {
12     float h = dht.readHumidity();
13     float t = dht.readTemperature();
14     Serial.print("Humidity: ");
15     Serial.print(h);
16     Serial.print(" %, Temperature: ");
17     Serial.print(t);
18     Serial.println(" °C");
19     delay(2000);
20 }
```

# PLC(Programmable Logic Controller)

PLC(Programmable Logic Controller)는 산업 자동화의 핵심 요소입니다

- PLC 하드웨어 구조 (CPU, 입출력 모듈, 통신 모듈)
- 래더 로직 프로그래밍
- 타이머와 카운터 활용
- 아날로그 신호 처리
- HMI(Human Machine Interface) 연동

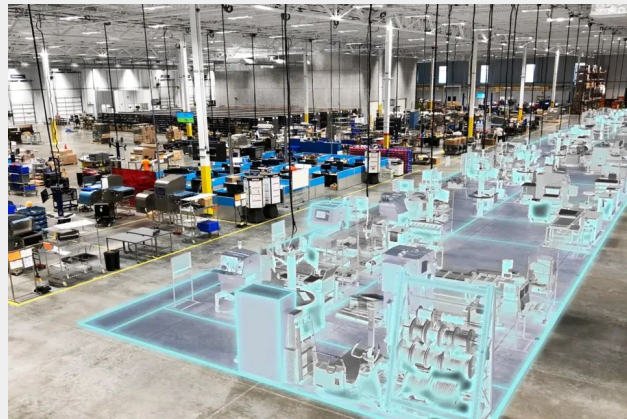
# PLC(Programmable Logic Controller)



# Unity 3D 엔진을 이용한 디지털 트윈 기술

## 주요 구현 단계

- 3D 모델링 (공장 설비, 로봇 등)
- 센서 데이터 연동 (REST API 또는 WebSocket 활용)
- 실시간 시각화 및 애니메이션
- UI/UX 설계 (데이터 대시보드, 제어 패널 등)



# Unity 3D 엔진을 이용한 디지털 트윈 기술



```
using System;

namespace MyProgram
{
    2 references
    class MyClass
    {
        private string myVariable;
        0 references
        static void Main()
        {
            MyClass myClass = new();    //new()는 new MyClass()를 줄인 표현
            myClass.MyMethod();
        }

        1 reference
        private void MyMethod()
        {
            myVariable = "Hello World!";
            Console.WriteLine(myVariable);
        }
    }
}
```

by Henry

# Realtime Database를 이용한 스마트 팩토리 구축

Firestore는 실시간 데이터베이스와 클라우드 함수를 제공하여 IoT 디바이스와 클라이언트 애플리케이션 간의 실시간 데이터 동기화를 가능하게 합니다.

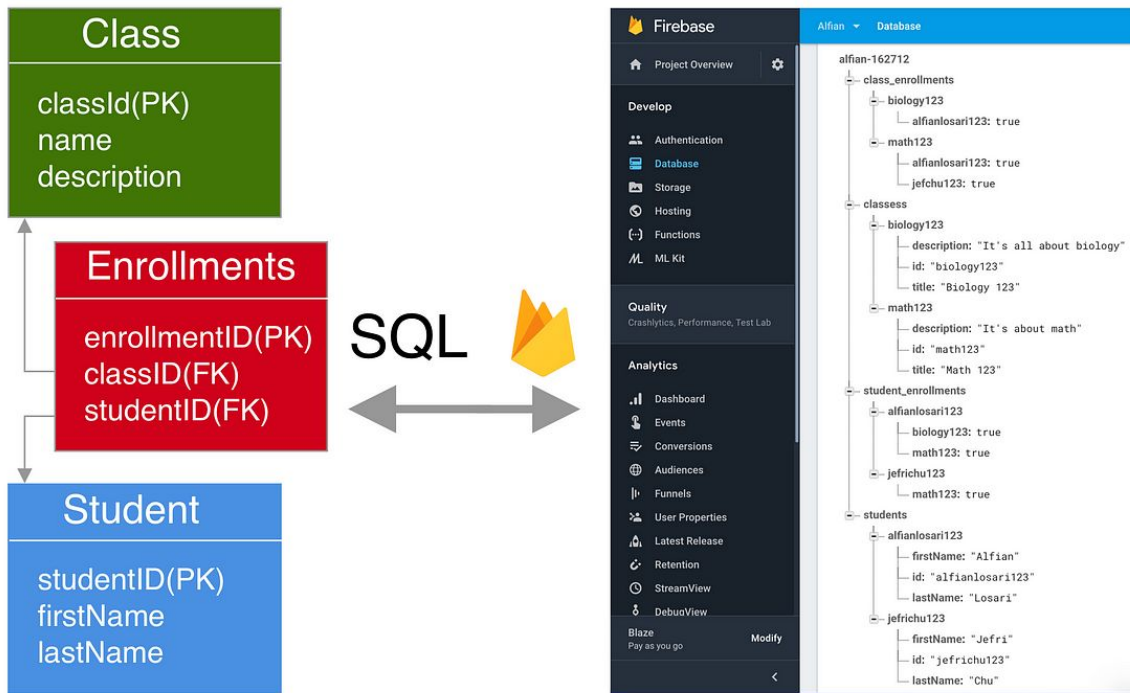
주요 구현 단계

- Firebase 프로젝트 설정
- 실시간 데이터 구조 설계
- 보안 규칙 설정
- 클라이언트 애플리케이션 개발 (웹/모바일)





# Realtime Database를 이용한 스마트 팩토리 구축



# 정리

이러한 기술들을 통합하여 스마트 팩토리를 구축하는 과정

1. 요구사항 분석 및 시스템 설계
2. 센서 네트워크 구축
3. PLC 시스템 구현 및 연동
4. 클라우드 인프라 구축 (Firebase)
5. 디지털 트윈 개발 (Unity 3D)
6. 시스템 통합 및 테스트
7. 모니터링 및 분석 시스템 개발
8. 최적화 및 유지보수

**Thank you**