

1. Implementation

Introduction

This chapter explains the actual development and integration of the proposed IIoT-based monitoring and maintenance system. The development was divided into three main layers: PLC programming, edge data acquisition, and the Django-based web application. All components have a core role in enabling real-time monitoring, fault reporting, and maintenance management.

1.1. PLC Programming (TIA Portal)

The Siemens S7-1200 PLC was programmed using TIA Portal to monitor and control a prototype sorting machine. The machine identifies and classifies parts based on height and color using sensors. The PLC also manages actuators (e.g., conveyors, reject mechanisms) and tracks machine status.

Key elements:

- Input signals from sensors detect part presence and type.
- Output signals drive actuators based on classification logic.
- Memory addresses (in Data Blocks) store:
 - Total part count
 - Number of parts by type
 - Current machine status (e.g., idle, sorting, error)
 - Error flags for specific faults (e.g., jammed part, sensor failure)

These values are continuously updated and made available for external read access by the edge device.

1.2. Edge Communication

1.2.1. Python Script

A Python script running on an edge device was developed to:

- Connect to the S7-1200 via Ethernet using the Snap7 library.
- Read critical data points from the PLC.
- Convert binary/byte data into human-readable values (integers, strings).
- Send the data to the Django web server using HTTP POST requests.

1.2.2. Script Snippet:

```
import snap7
import requests

plc = snap7.client.Client()
plc.connect('192.168.0.1', 0, 1) # Adjust IP, rack,
and slot
```

```
while True:
    part_count = plc.db_read(1, 0, 2)
    status = plc.db_read(1, 2, 2)

    data = {
        "machine_id": "sorter01",
        "part_count": int.from_bytes(part_count,
'big'),
        "status_code": int.from_bytes(status, 'big')
    }

    requests.post("http://your_web_app/api/machine-
data/", json=data)
```

1.2.3. Node-RED

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequi doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplifi- carique non possit. At etiam Athenis, ut e.

1.3. Web Application

1.4. REST API Structure

Table 1: API structure

Endpoint	Method	Description
/api/machine-data/	POST	Receives machine data from PLC

1.4.1. Subsection 2.2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequi doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplifi- carique non possit. At etiam Athenis, ut e.



Figure 1: Typst logo

Figure 1 shows the Typst logo.

Conclusion

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequi doleamus animo, cum corpore dolemus, fieri tamen permagna.