Internet of Things

A.Y. 2023/2024

System for **C**ollection, **R**ecycling, and **A**utomated **P**rocessing

Repository: **https://github.com/giadasql/IoT_Project**

Beccari Giada

Table of Contents

Introduction

SCRAP (System for Collection, Recycling, and Automated Processing) is an IoT system designed to monitor and automate the use of recycling centers. Normally, this service requires the presence of an operator on-site, who manually weighs and records the amount of waste deposited by each user into a software system. By automating this process, this project aims at providing a faster and more efficient service, potentially available at all times without on-site supervision.

The system equips each bin with various sensors to track its status and contents. Citizens would use a personalized card to access the center, dispose of their waste independently, and leave without needing assistance. The system automatically measures the deposited waste and stores the data in the cloud. This enables detailed analysis of waste production within a municipality and allows tracking of individual recycling contributions.

The system could also help automate some aspects of bin maintenance. For example, if a bin is equipped with a compactor, it could be activated automatically when the waste reaches a threshold level.

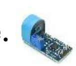
A dashboard provides real-time monitoring of one or more centers, providing an overview of the status of each bin, and allowing remote manual operations.

User Interaction Process:

1. Approach the bin designated for the waste material and scan an RFID card at the reader.
2. The system unlocks the bin, and the lid opens.
3. Deposit the waste.
4. Remove the RFID card, which automatically closes the lid and ends the session.
5. The system records sensor data and logs a transaction, registering the waste weight in kilograms.

Components

In the recycling center, there are one or more bins for each material. Each waste bin is equipped with the following devices:

- **RFID reader** that residents use to activate the bin by scanning their personal card. 
- A **magnetic sensor** to detect whether the lid is open or closed. 
- A **LIDAR sensor** to measure how full the bin is. 
- A **current sensor** to check if the compactor is active. 
- A **scale** to measure the weight of waste inside the bin.
- A **lid actuator** to open and close the lid.
- A **compactor actuator** to turn the compactor on and off.

Each bin has a **collector device** that gathers data from all the sensors and forwards it to an external application. The collector identifies all connected devices upon startup via a configuration message.

To allow communication with services that are hosted outside of the local network, one **RPL border router** is also installed.

There are two external applications:

- A **Cloud Application** which receives and stores all the data coming from one or more recycling centers.

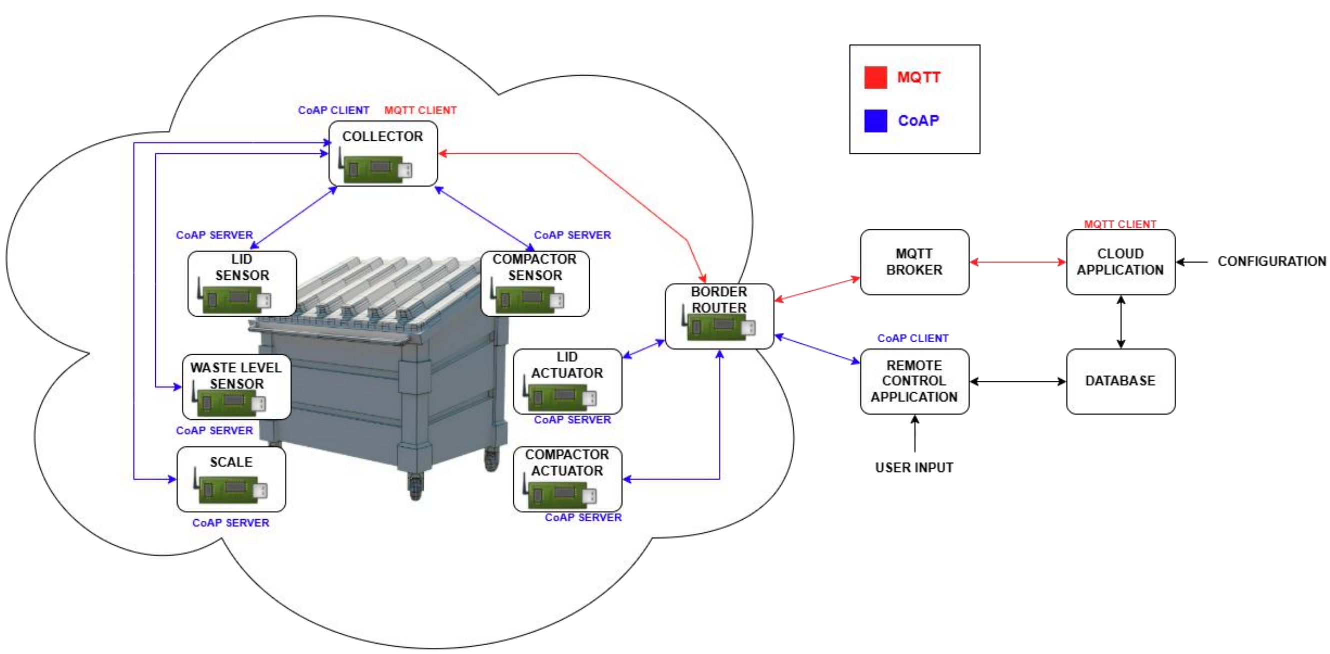
- A **Remote Control Application** uses the data stored by the Cloud Application to perform automatic and manual operations.

Architecture

Communication inside and outside the WSN happens via MQTT and CoAP. In particular:

- Each **sensor** and **actuator** in the Wireless Sensor Network (WSN) functions as a **CoAP server** exposing one resource, which corresponds to the sensed value or the actuator command.
- The **collector** device acts as **CoAP client** and **MQTT client**.
- The **Cloud Application** acts as **MQTT client** in order to receive the live data from the collector devices.
- The **Remote Control Application** acts as **CoAP client** in order to send commands to the devices in the WSN.

Below is a representation of the whole system, considering only a single bin for simplicity.



Implementation

Each sensor in the WSN is a CoAP server that exposes one endpoint allowing **GET** operations without parameters.

Sensor Endpoints:

- **Compactor Sensor:** */compactor/active* (Boolean: true = active, false = inactive)
- **Lid Sensor:** */lid/open* (Boolean: true = open, false = closed)
- **RFID Reader:** */rfid/value* (String: user ID if present, "No data" otherwise)
- **Weighing Scale:** */scale/value* (Float: weight in kg)
- **Waste Level Sensor:** */waste/level* (Integer: percentage of bin capacity used)

For simulation purposes, all these endpoints also allow **PUT** operations. This would not be part of a real system, but it is necessary for this project to easily change the values and simulate a real interaction with the system.

The actuators are CoAP servers as well, they expose one endpoint allowing **PUT** operations. This allows them to receive commands from remote applications.

Actuator Endpoints:

- **Compactor Actuator:** */compactor/command* (accepts: "turn on", "turn off")
- **Lid Actuator:** */lid/command* (accepts: "open", "close")

Each actuator also includes a **physical button** for manual operation.

For simulation reasons, the two actuators also expose an additional endpoint which is used to configure the addresses of the two related sensors (the compactor active sensor and the lid sensor). This is necessary in the project to simulate the real system. When the lid actuator is pressed (or commanded remotely) it must update the value sensed by the lid sensor. The same happens in the case of the compactor actuator and sensor.

The **collector device** is a CoAP client as well as an MQTT client. As a CoAP client, it sends periodic requests to read the data from all the sensors located in the same bin.

As an MQTT client, it publishes on the topics:

- “**config/request**”: used to send messages in the following format:

  The collector sends its own address. This message is sent when the collector first connects to the MQTT broker and is sent at regular intervals until a valid response is received.


```
{
  "collector_address": "fe80::x:x:x:x"
}
```


- “**bins**”: used to send messages with the following format:


```
{
  "collector_address": "fe80::x:x:x:x",
  "lid_sensor": "closed",
  "compactor_sensor": "off",
  "scale": 15.5,
  "waste_level_sensor": 20,
  "rfid": "No data"
}
```


  The message describes the status of one specific bin and is sent periodically.

And it subscribes to the topic:

- “**config/response**”: responses to configuration requests are sent on this topic by an external application. Configuration response messages have the following format:


```
{
  "collector_address": "fe80::x:x:x:x",
  "lid_sensor_address": "fe80::x:x:x:x",
  "compactor_sensor_address": "fe80::x:x:x:x",
  "scale_address": "fe80::x:x:x:x",
  "waste_level_sensor_address": "fe80::x:x:x:x",
  "rfid_reader_address": "fe80::x:x:x:x"
}
```

All the message bodies are in **JSON** format because it is more compact than XML, leading to smaller messages which makes communication more efficient.

Finally, the **RPL border router** is responsible for connecting the devices in the WSN with the rest of the internet.

The **Cloud Application** is responsible for handling configuration requests. The application exploits a XML file which contains the list of devices associated with each collector in the system. The file follows the format:

```xml
<bins>
    <bin id="bin01">
        <collector_address>fe80::f6ce:367b:9c44:e26a</collector_address>
        <lid_sensor_address>fe80::f6ce:36aa:8370:9738</lid_sensor_address>
        <lid_actuator_address>fe80::f6ce:36fc:e563:b772</lid_actuator_address>
        <compactor_sensor_address>fe80::f6ce:36dc:2477:3d6d</compactor_sensor_address>
        <scale_sensor_address>fe80::f6ce:36d4:b744:3ef3</scale_sensor_address>
        <waste_level_sensor_address>fe80::f6ce:36be:c885:2c83</waste_level_sensor_address>
        <compactor_actuator_address>fe80::f6ce:3620:b4ba:4310</compactor_actuator_address>
    </bin>
</bins>
```

The application is also responsible for collecting the bin status data and storing it in a MySQL database. To reduce the amount of storage required, the application stores a new value for a sensor only when it is different from the value received in the previous message.

The **MySQL database** has the following tables:

	bins_change_log
PK	log_id int NOT NULL
	bin_id varchar(255)
	change_timestamp timestamp
	new_value varchar(255)
	sensor_name varchar(255)

	bins_rfid_transactions
PK	transaction_id int NOT NULL
	bin_id varchar(255) NOT NULL
	rfid varchar(255) NOT NULL
	start_time timestamp
	end_time timestamp
	weight_difference decimal

	bins_current_state
PK	bin_id varchar(255) NOT NULL
	last_updated varchar(255)
	rfid varchar(255)
	compactor_state varchar(255)
	lid_state varchar(255)
	scale_weight decimal
	waste_level decimal

	alarms
PK	alarm_id int NOT NULL
	time timestamp
	bin_id varchar(255)
	message varchar(255)
	acknowledged boolean

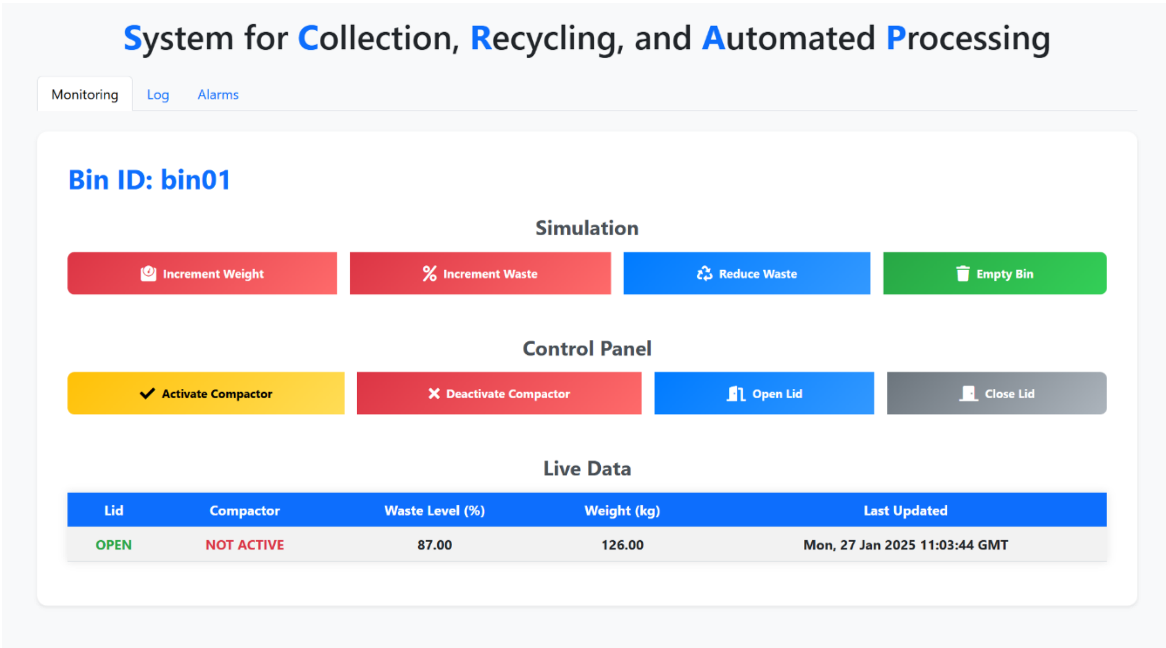
The **Remote Control Application** has three main functions:

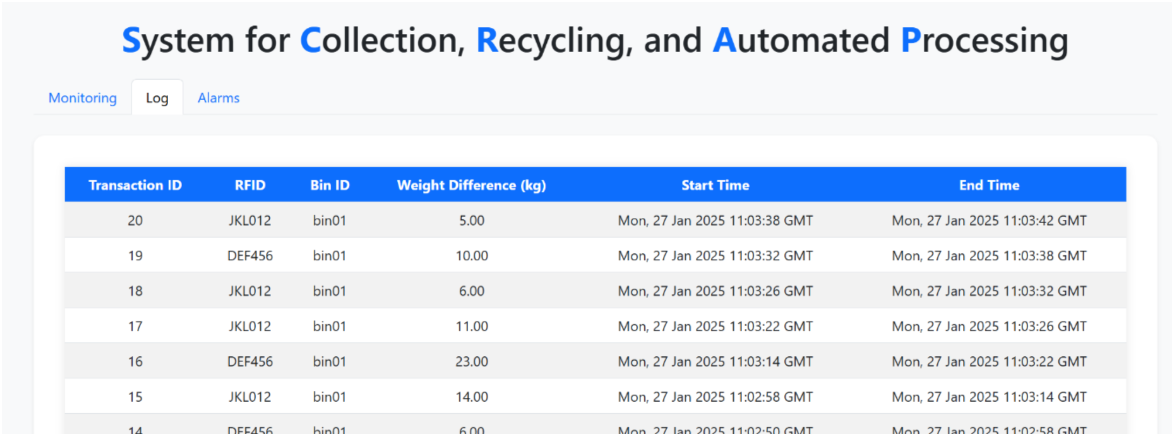
- **Automatic Operations:** automatically activates the compactor when the bin is full above a configured threshold. If the bin is still too full after the compactor has stopped, the application generates an alarm on the database. Alarms can then be seen and acknowledged on the GUI.

- **Real-time Monitoring:** It provides a monitoring GUI through which operators can see real-time information about each bin in the system. The interface shows, for each bin, the following data: code read by the RFID reader, status of the lid, status of the compactor, weight measured by the scale, waste level inside the bin.
  On another tab, the operator can see a table with all the transactions done by users. A transaction is composed of the user's id, the start and end times, the weight of the deposited waste, the bin id.

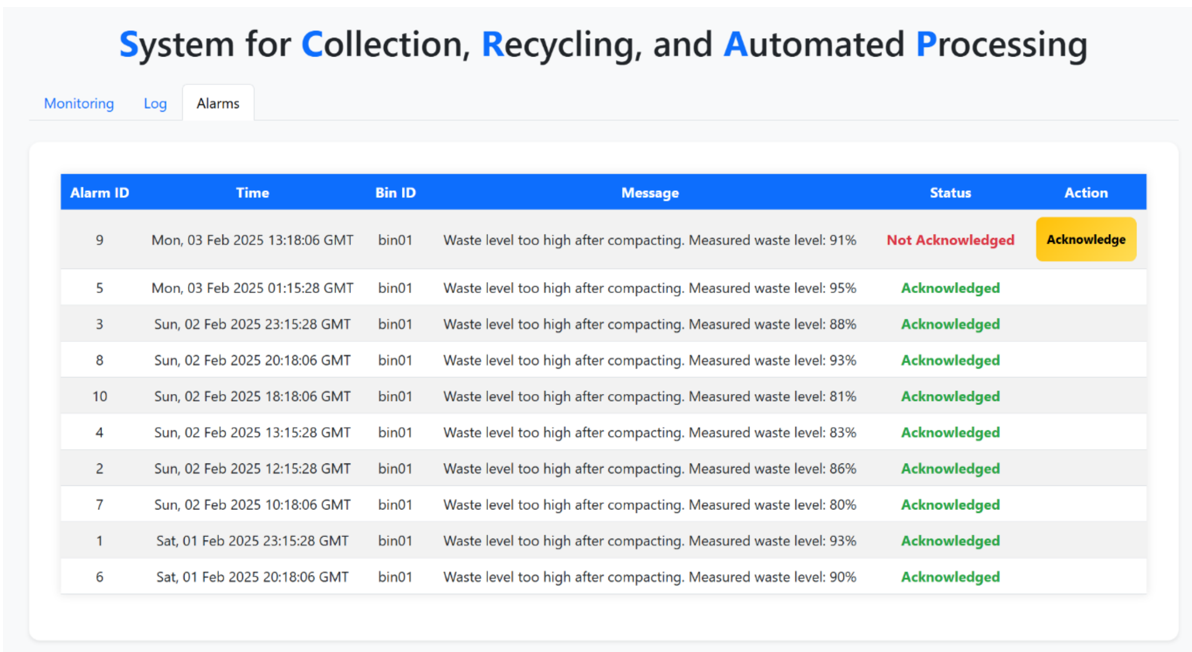
- **Remote Commands:** It provides a command interface that the operator can use to interact remotely with a bin. The operator can open and close the lid, and turn the compactor on and off.

For simulation reasons, this application also includes a control panel with some buttons to manipulate the status of the system.


System for Collection, Recycling, and Automated Processing
Monitoring
Log
Alarms
Bin ID: bin01
Simulation
Increment Weight
Increment Waste
Reduce Waste
Empty Bin
Control Panel
Activate Compactor
Deactivate Compactor
Open Lid
Close Lid
Live Data
Lid
Compactor
Waste Level (%)
Weight (kg)
Last Updated
OPEN
NOT ACTIVE
87.00
126.00
Mon, 27 Jan 2025 11:03:44 GMT



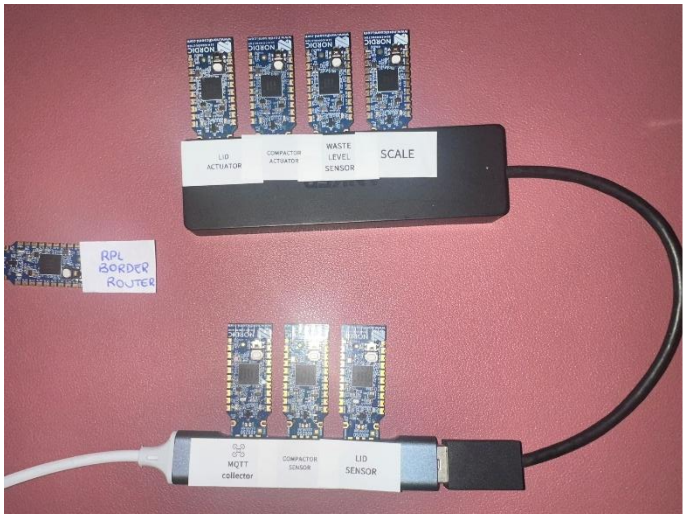
Transaction ID	RFID	Bin ID	Weight Difference (kg)	Start Time	End Time
20	JKL012	bin01	5.00	Mon, 27 Jan 2025 11:03:38 GMT	Mon, 27 Jan 2025 11:03:42 GMT
19	DEF456	bin01	10.00	Mon, 27 Jan 2025 11:03:32 GMT	Mon, 27 Jan 2025 11:03:38 GMT
18	JKL012	bin01	6.00	Mon, 27 Jan 2025 11:03:26 GMT	Mon, 27 Jan 2025 11:03:32 GMT
17	JKL012	bin01	11.00	Mon, 27 Jan 2025 11:03:22 GMT	Mon, 27 Jan 2025 11:03:26 GMT
16	DEF456	bin01	23.00	Mon, 27 Jan 2025 11:03:14 GMT	Mon, 27 Jan 2025 11:03:22 GMT
15	JKL012	bin01	14.00	Mon, 27 Jan 2025 11:02:58 GMT	Mon, 27 Jan 2025 11:03:14 GMT
14	DEF456	bin01	6.00	Mon, 27 Jan 2025 11:02:50 GMT	Mon, 27 Jan 2025 11:02:58 GMT



Alarm ID	Time	Bin ID	Message	Status	Action
9	Mon, 03 Feb 2025 13:18:06 GMT	bin01	Waste level too high after compacting. Measured waste level: 91%	Not Acknowledged	Acknowledge
5	Mon, 03 Feb 2025 01:15:28 GMT	bin01	Waste level too high after compacting. Measured waste level: 95%	Acknowledged	
3	Sun, 02 Feb 2025 23:15:28 GMT	bin01	Waste level too high after compacting. Measured waste level: 88%	Acknowledged	
8	Sun, 02 Feb 2025 20:18:06 GMT	bin01	Waste level too high after compacting. Measured waste level: 93%	Acknowledged	
10	Sun, 02 Feb 2025 18:18:06 GMT	bin01	Waste level too high after compacting. Measured waste level: 81%	Acknowledged	
4	Sun, 02 Feb 2025 13:15:28 GMT	bin01	Waste level too high after compacting. Measured waste level: 83%	Acknowledged	
2	Sun, 02 Feb 2025 12:15:28 GMT	bin01	Waste level too high after compacting. Measured waste level: 86%	Acknowledged	
7	Sun, 02 Feb 2025 10:18:06 GMT	bin01	Waste level too high after compacting. Measured waste level: 80%	Acknowledged	
1	Sat, 01 Feb 2025 23:15:28 GMT	bin01	Waste level too high after compacting. Measured waste level: 93%	Acknowledged	
6	Sat, 01 Feb 2025 20:18:06 GMT	bin01	Waste level too high after compacting. Measured waste level: 90%	Acknowledged	

Deployment

All the sensors and actuators, the collector and the border router are **Nordic Semiconductor nRF52840 Dongle** boards. Each one of them runs a different program based on the Contiki operating system.



Some of the devices exploit the led and the button present on the board:

- The lid sensor uses the led to indicate the state of the lid. **Green** corresponds to **open**, and **red** corresponds to **closed**.
- The compactor sensor uses the led to indicate whether the compactor is running. The led is **red** when the compactor is **active** and turned **off** when it's **inactive**.
- The lid actuator uses the button to open and close the lid. Pressing the button toggles the state.
- The compactor actuator uses the button to activate and deactivate the compactor in a similar way.