2019

Universitetet i Sørøst-Norge

Autonomous waste sorter

YASIN YARI
222164

Table of Contents

1 Introduction

In the modern world, garbage has become a prominent problem. Studies show that most of the waste have used only once [1]; therefore, there is an increasing need for raw material for production. One way to reduce the garbage and counteract with the elevating need for the raw material issues is to recycle the trash we dispose of. In order to do recycling, the wastes should be sorted. Nowadays, using different containers for different types of trash is common, but there is a possibility of human error. The mixed-type trash in a container makes the recycling process more difficult and costly because a further separation process is needed. By making an intelligent recycle bin which can sort waste autonomously, it is possible to avoid human errors and as a result, increase recycling and reproduction pace. The system is using Atmel 2560 microprocessor. This report is made for safety critical systems (ES-SCS5300). This documentation is containing various specifications of the system with focus on safety aspects of the system.

1.1 System Overview

The system has a main container in which users can throw the trash in it. By using both software and hardware tools, the trash will be detected, identified, and classified. Then the recognized trash will be led to a proper sub-container by a mechanical tool.

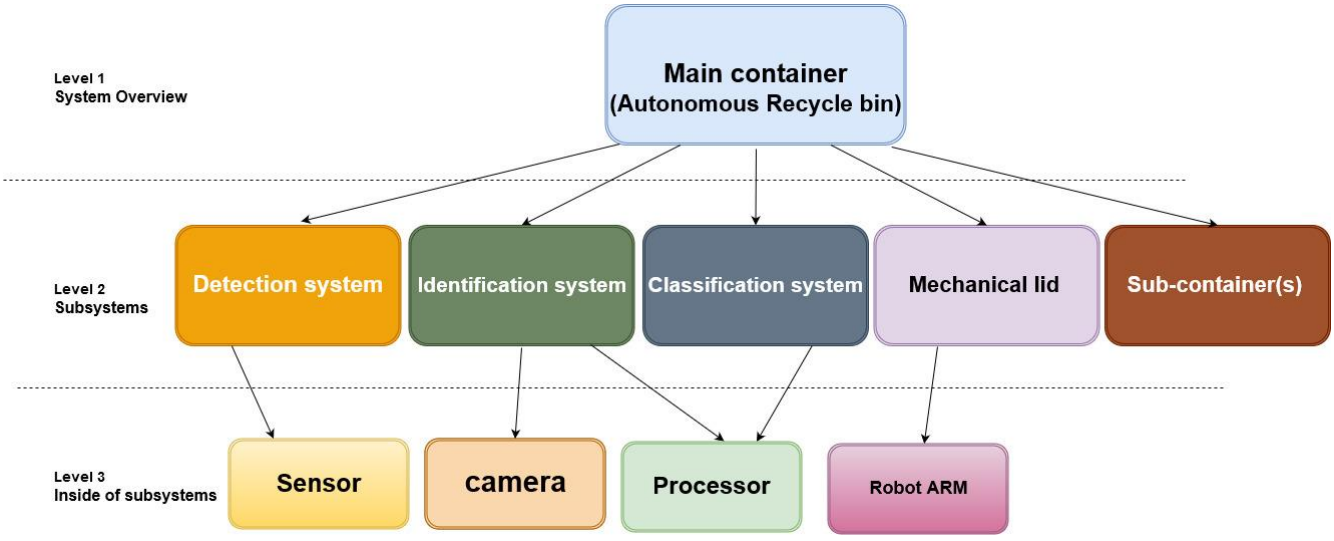


Figure 1. Overview of the system

As can be seen, the system has a main container which includes all the subsystems. The subsystems are Sensors for both object detection and recognition, a mechanical arm that can move the garbage, various sub-containers for different types of trash, and a processor as brain of the systems which sends order to other components.

1.2 Data Flow

There is a main container which user throws trash in it. After receiving trash, the trash will be scanned by camera and sensors. While the trash will be led to the mechanical arm, the data of the

scanned trash will be sent to the processor. The processor identifies the trash type by using a machine learning algorithm and a proper data set. The processor decides which sub-container the identified trash should go to. The processor sends the command to the mechanical tool to put the identified trash in the proper sub-container. The mechanical tool will do the order of the processor.

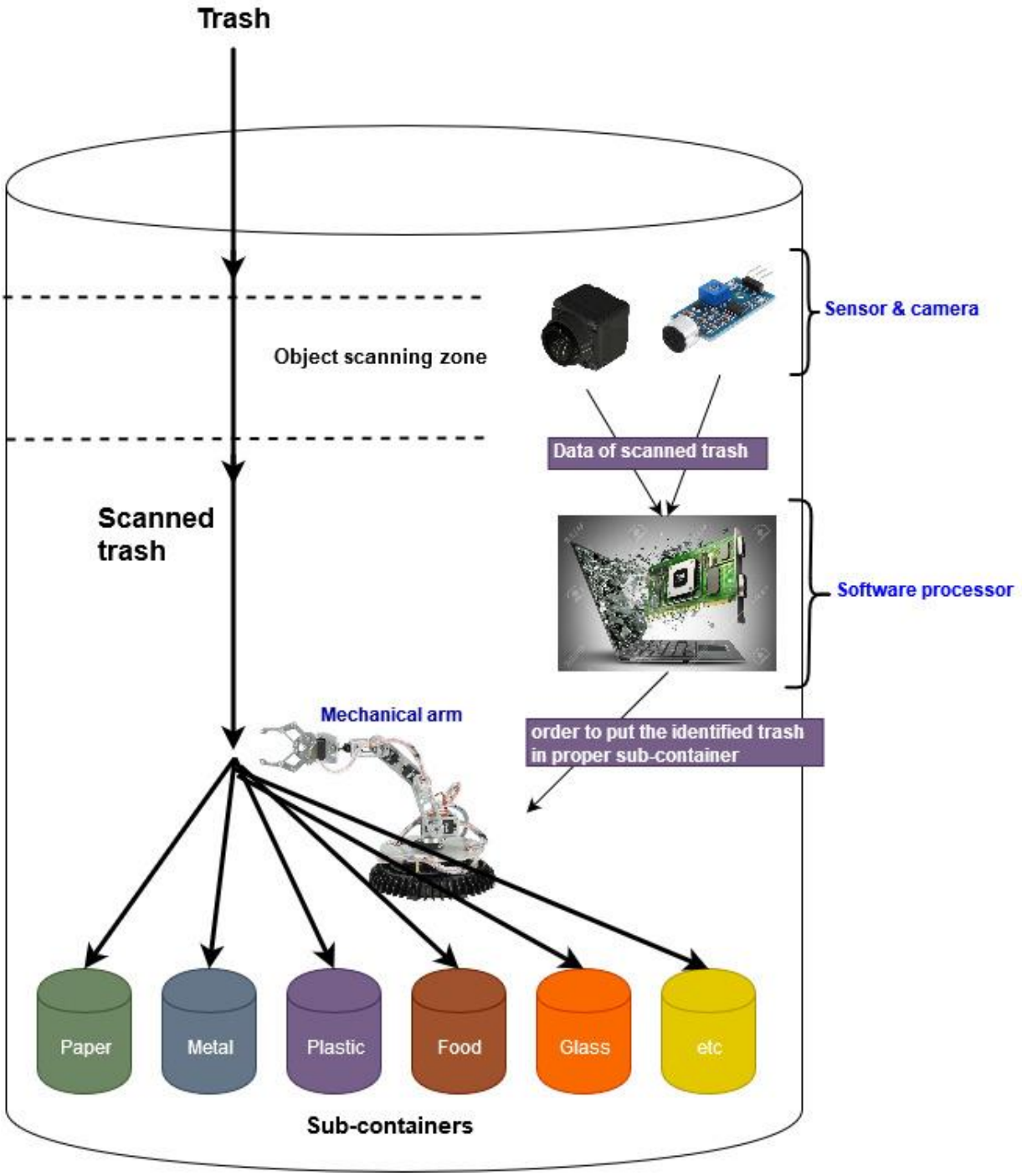


Figure 2. High level design of the system

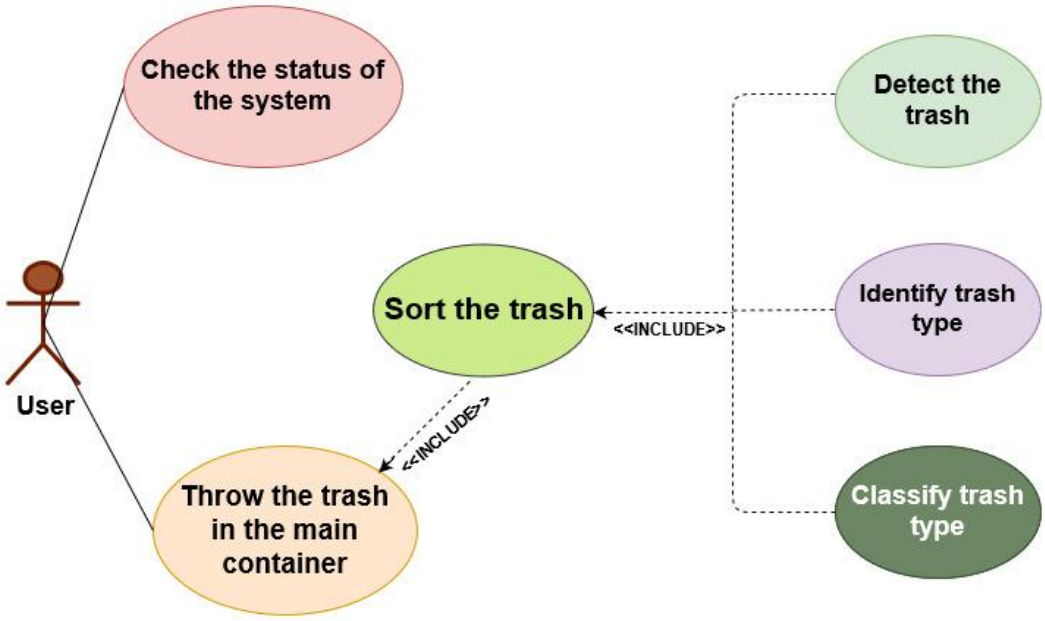


Figure 3 Use Case Diagram

1.3 Project Overview

Among all various available life cycles, it is decided to use V model. The V model has top-down approach in the first half of the cycle and Bottom- up approach for the second half which can cover the system. It shows the plan of after finishing each phase of the life cycle. Testing and designing occurs before coding that saves a lot of time for coding. I am also following safety lifecycle model given below which provides a step by step process for safety of a system.

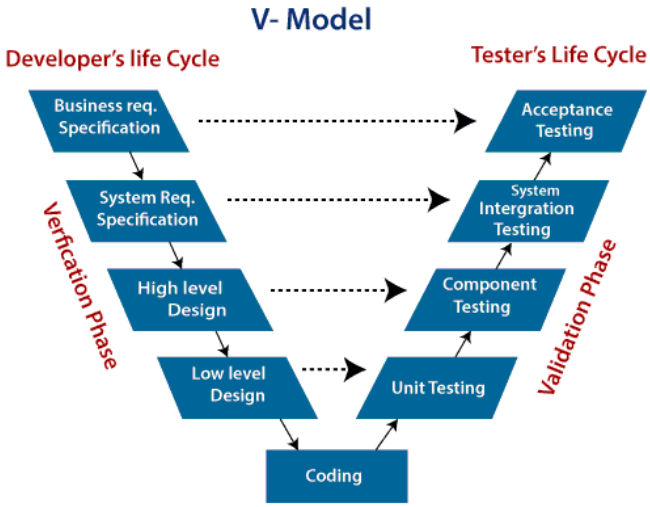


Figure 4. V model overview

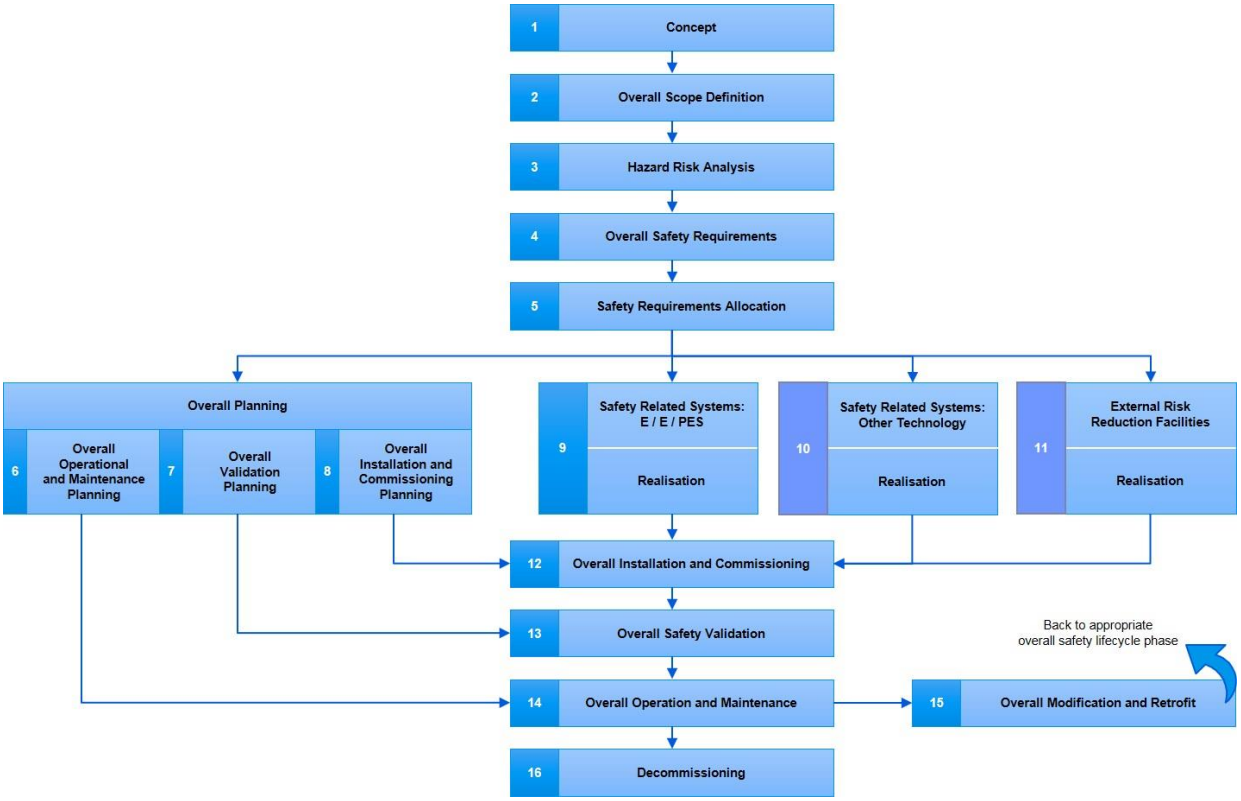


Figure 5 Overall safety Life-cycle Model [2]

2 Requirement Specification

Each system has its own requirement based on stakeholders demands. The final system must cover the requirements of the clients to be approved by them.

2.1 Functional Requirements

ID	Functional Requirement
FR11	The system must detect common types of trash.
FR12	The system must detect trash within 0.5 second.
FR13	The system must have processing time of 1-2 second in order to decrease latency.
FR14	The system must be able to notify users about it's status. (User Interface)
FR15	The system must function entirely autonomous to avoid human errors.
FR16	The system must follow safety standard of SCSC.
FR17	The system must have at least 90% accuracy for sorting.

2.2 Safety requirements

ID	Hazard	Safety Requirement
SR11	H3	The system must not misinterpret the trash type.
SR12	H5	The system must not cause any chemical operation for garbage.
SR13	H2	The sub-containers should be fire-proof in case of any fire in the using environment.
SR14	H4, H2	The system must not cause any electrical shock for the user.
SR15	H2, H5	The system must be entirely safe when using by children.

2.3 None-functional requirements

ID	Non-Functional Requirement
NF1	Ease of maintenance
NF2	Cost efficiency
NF3	Power efficiency

2.4 Reliability Requirements

ID	Reliability Requirement
RR11	In case of an unknown input or unrecognize trash, system should not mix it with other trash and should be able to sort it in a separate container as unknown. **System Fault Tolerance**
RR12	System should be able to sort very small pieces of trash. **System maturity**
RR113	The software part of the system should not have any run time error or infinite loops. In case of happening, there must be exception handling mechanism. (Watchdog reset). **System recoverability**
RR14	System should be able to handle mixed-type trash. **System Fault Tolerance**
RR15	The system should contain a large data set of different types of trash. **System Maturity**

System Maturity	❖ Error to handle input ❖ Error to produce output ❖ Error to produce correct output
System recoverability	❖ Failure operations ❖ Failure Mechanism
System fault tolerance	❖ Fault detection ❖ Fault removal ❖ Fault prevention

3 Hazard Analysis

In order to do hazard analysis, we use fault tree, HAZOP (hazard and operability analysis), and FMEA (failure mode and effects analysis).

3.1 Fault tree analysis

FTA is combination of events and logic gates to analyze the reason of undesired events or state of the system.

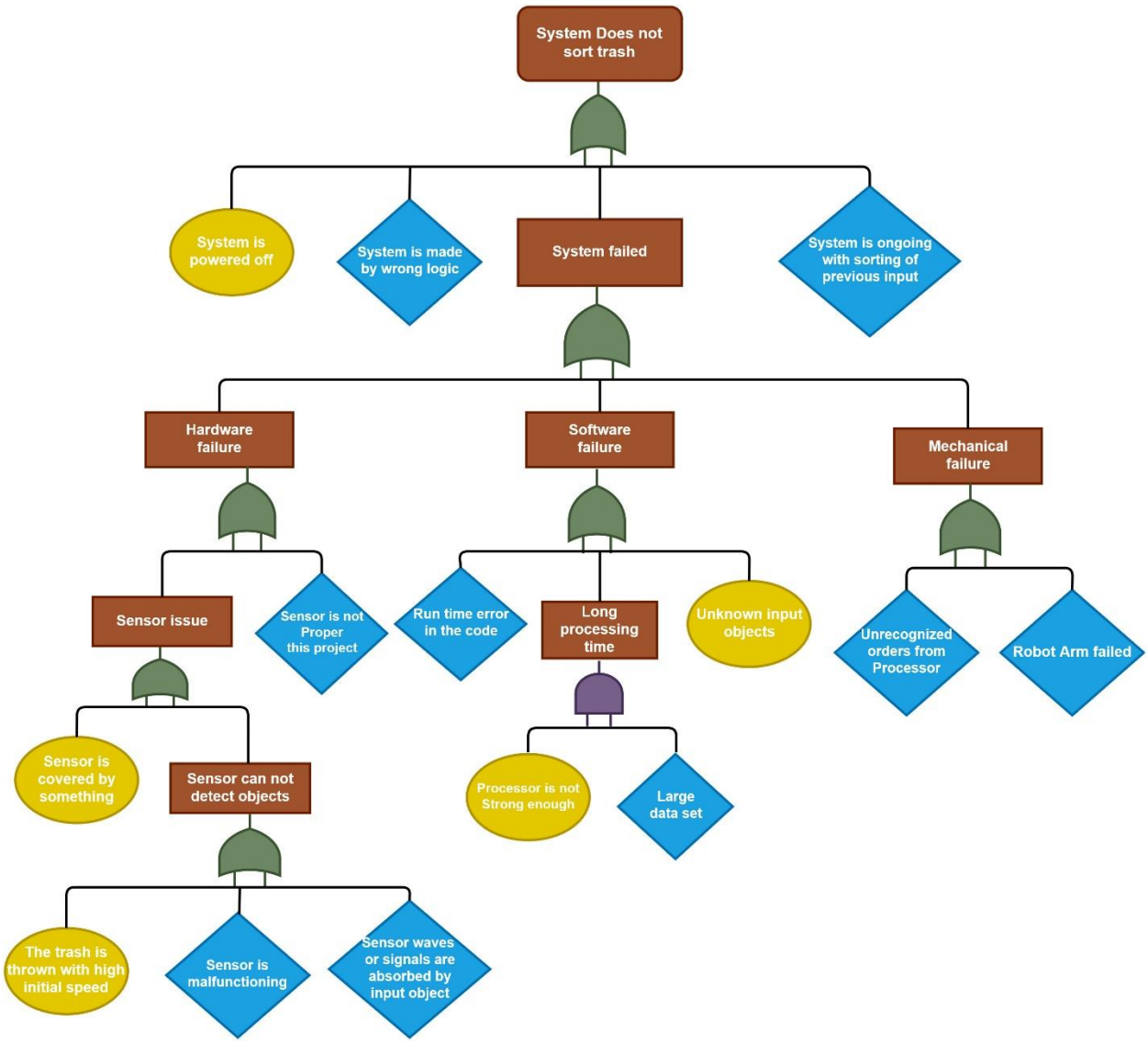

System Does not sort trash
System is powered off
System is made by wrong logic
System failed
System is ongoing with sorting of previous input
Hardware failure
Software failure
Mechanical failure
Sensor issue
Sensor is not Proper this project
Run time error in the code
Long processing time
Unknown input objects
Unrecognized orders from Processor
Robot Arm failed
Sensor is covered by something
Sensor can not detect objects
Processor is not Strong enough
Large data set
The trash is thrown with high initial speed
Sensor is malfunctioning
Sensor waves or signals are absorbed by input object

Figure 6 FTA

3.2 Hazards

After doing the hazard analysis, the hazards are listed.

ID	Hazard	Input	Output	Consequences
H1	If someone threw a trash with high initial velocity, the scanners cannot scan it in a short time.	Trash with high velocity	Unrecognized trash	Sorting fails
H2	If someone throw a flammable trash, the system might get fired.	Flammable trash	Fire in the system	Damage to the system
H3	If system receives a mixed-typed trash, System cannot classify it properly.	Mixed-typed trash	Trash recognized as one type depend on which side is front of scanners	Wrong sorting
H4	If someone throws a moist or wet trash or the using environment is moist, it may damage the (or rain in open space usage) electronics	Moist trash Moist environment	Moist environment for electronics	Damage to the electronics
H5	If system receives a toxic trash, system should not spread it to other trashes or the using environment.	Toxic trash	Contaminated environment	Spreading disease
H6	Someone putting his/her hand inside the autonomous recycle bin	Human body	-	Damaging electronics Electrical shock
H7	If there is any fluctuation in power electricity voltage, system may damage	Trash	Unsorted Trash	Sorting fails

SEVERITY OF CONSEQUENCES		
Aviation definition	**Meaning**	**Value**
Catastrophic	Results in an accident, death or equipment destroyed	5
Hazardous	Serious injury or major equipment damage	4
Major	Serious incident or injury	3
Minor	Results in a minor incident	2
Negligible	Nuisance of little consequence	1

4 Risk Analysis

We identified the hazards in the previous section. Exposure on hazard causes risk for the system.

Id	Related Hazard	Risk	Severity	probability	Impact	Prevention Method
R1	H1	Receiving trash with high velocity	Moderate	Likely	Critical	Using a middle layer that moves off after detection of trash
R2	H2	System getting fire due to flammable trash or components and wiring issue	Critical	Ocasional	Critical	Using fire resistant material- (to diminish the side effect in case of happening this risk, we can mount a fire alarm close to system)
R3	H3	Getting mixed-type trash	Moderate	Likely	Marginal	Having a separate container for unknown trash
R4	H4	Getting moist trash	Neglible	Ocational	Negligble	Electronics are isolated and mounted in a proper location
R5	H5	Getting toxic trash	Critical	Seldom	Critical	Forbid the use of device for such a trash
R6	H6	A human takes his/her hand in the system.	Critical	Seldom	Critical	Make sure of wiring and mount electronics in a safe position

4.1 HAZOP

No	Unit	Req	Guide work	Deviation	Consequence	Cause	Action
1	Sensor	FR12	Not Detecting objects	Sensor does not detect the trash.	The trash won't be sorted properly	Sensor has sensing issue. A trash arrived with high initial speed. The signals and waves are absorbed by input object.	Calibration. Reconfiguration.
2	Sensor	FR12	Not Recognize objects	Sensor cannot recognize two types of trash at same time.	If there is a mixed-type trash, it won't be sorted properly.	Sensor can scan one object per attempt.	Having a separate sub container for mixed-type trash. Using multiple sensors.
3	Sensor	FR12	No input	Sensor detects object while there is no input.	The system is in busy mode and won't be able to have new input.	Sensor is covered by something or it is malfunctioning.	Having multiple sensors. More often maintenance.
4	Sensor	FR12	No data sending	Sensor won't be able to send the scanned data to the processor.	Sorting fails.	The is some technical issue in the signal transmission.	Reconfiguration.
5	Processor (Control system)	FR11	No match	Processor is not able to find any matches between scanned data and data set.	The sorting fails.	The data set is not large enough to cover all types of trash.	Update the data set. Having a separate sub container for unrecognized trash.
6	Processor (Control system)	FR13	Long	Processor takes too long to recognize trash type and send proper order to mechanical arm	High latency	The data set is very large. The processor (GPU/CPU) is not fast enough.	Having stronger processor. Decrease the volume of unimportant data in data set.
7	Processor (control system)	FR15	No send	Processor cannot send proper order to the mechanical arm.	Sorting fails.	Run time error. Technical issue.	Regular maintenance. Efficient coding.

8	Processor (control system)	FR15	Unproper order	Processor is not ordering properly to the other subsystems as control unit	Sorting fails	The issue in the logic of the whole system.	Investigate the logical issues in the system.
9	Mechanical Arm	FR15	Not working	The mechanical arm is not working properly.	Sorting fails	Technical failure. Mechanical issue	Regular checking
10	The entire system	FR13	No respond	System is not able to respond to the inputs fast enough.	Sorting fails	Having too many input trashes together.	Warn the user whether the system is ready to use or not.
11	The entire system	FR15	No cooperation	The subsystems are not cooperating properly.	Sorting fails	Integration problem.	Having proper bottom-up integration and testing.
12	The entire system	FR16	material	The system is not made by anti-fire materials.	Damage to the system	Flammable trash	Using anti fire material for making the system and container.
13	The entire system	FR16	Causing issue	The system is causing chemical operation or not preventing the smell of garbage's from spreading to the using environment	Contamination of the using environment	Some trashes may cause chemical operation. Some trashes may smell after decaying.	Proper soring accuracy and regular maintenance.

4.2 FMEA

This process comprises evaluation of as many subsystem and component as possible to find out every potential failure is the system, together with cause of that failure.

No	Req	Unit	Failure mode	Possible cause	Local effect	System Effects	Remedial action
1	FR11	Sensor	Not working properly	Physical damage Waves absorption	Fail to scan inputs	Sorting fails.	Using multiple sensor
2	FR15	Sensor	Latency in detection	Old version Bad positioning	Fail to scan inputs	Sorting fails	Using newest sensors Using multiple sensor
3	FR15	Processor (Control system)	Low processing speed	Large data set Weak processor	Fail to send proper orders to other components (subsystems)	Sorting fails	Hire strong GPU Remove unuseful information from data set
4	FR16	Processor (Control system)	Not working (hardware)	Physical damage	Fail to process and control other subsystems	Sorting fails	Routine check
5	FR16	Processor (Control system)	Bug in Software (failure)	Run time error	System stops	Sorting fails	Testing Quality assurance
6	FR16	Mechanical arm	Not working	Physical damage	The classified trash won't be sorted	Sorting fails	Having a backup mechanical arm Routine check

5 Safety critical Measure for Software

5.1 Programming language selection

After evaluation of different languages, C++ and Embedded C are decided to use for development of the autonomous recycle bin system. The rationale of choosing C++ is that it is low level, faster than C and Python. C++ has higher abstraction level. C++ has closest access to hardware after embedded C. In small apps, C++ is very fast. So far, the detection part of the system is coded. It is written in Embedded C.

Since C is a high-level language it is not well suited to accessing low-level I/O ports or supporting interrupt vectors and ISRs. Embedded C is an extension of standard C, designed to work closer to the hardware, needing to access resources usually controlled by an operating system.

The key areas that have been added are:

- Fixed point math (fractional or integer + fraction, with size and saturated math variations) – reduces the memory-intensity and CPU cycle intensity of computations, necessary for low / medium powered μCs (as opposed to μPs).

- Multiple address spaces to accommodate mixed memories (e.g. RAM and FLASH) on μCs – necessary for application logic to directly reference memory locations and registers.

- IO register mapping (e.g. access to a USART register) – necessary for application logic to perform direct I/O and to communicate with I/O devices.

Embedded C is therefore ideal for embedded systems development because:

- It shares syntax and most of language with C, which is universally known (and standardized),
- It is very powerful and flexible,
- It converts efficiently into machine code (better than most high-level languages).

Development in a high-level language is generally much faster and easier than in assembly language, but assembly language usually gives more precise control, especially when accessing low-level resources such as registers, or where timing critical operation is necessary. Embedded C has the very powerful feature of being able to embed assembly code sections, for performance critical functions.

The safety measures which is considered for the code are:

1. “If” statement must have “else “. Also, the body must be compound statement.
2. Be careful when using dynamic memory.
3. Each “switch-case” statement needs to have “break” in the end.

4. “While” loop need loop counter to avoid infinite loop.
5. If the code is portable, the type of variables is important. For example, “int” might not be 32 bits in each complier.

5.2 Software design principles and tools selection

Software should be able to manage the sensors and processors and Robot Arm. It needs to handle exceptions properly. The code for detection is written in Atmel studio 7.0.

5.3 Design validation methodology

The software is being checked by simulation and using different input and evaluating the output with expected output. Although the software checking is important, the software should be check while its interacting with hardware. I have done the final validation of the system in section 7.

6 Safety critical measure for Hardware

6.1 Selection of components

6.1.1 Detection and Scanning tools

A lot of sensors are available for scanning the object that each of them are using different methods for that. Following the fact that the sensor is going to be used inside a main container, there is dark inside the recycle bin. So, the sensor cannot be a usual camera. I decided to use ultrasonic which works with sound waves to trigger not only the object identifier sensor but also the whole system. When an input arrives, Ultrasonic will detect the object and triggers the system. A light will light up the container and camera will scan the input object. ultrasonic is very cheap and enable us to use a normal camera which is also cheap. The latency of ultrasonic is 0.1 second which is more than enough.

6.1.2 Processor

As for the processor and control unit, ATMEL 2560 is chosen to interface with sensors. It’s not very strong processors for image processing. So, a separate GPU will also be added to the system to make hardware accelerations and decrease the latency and increase throughput.

6.1.3 Robot Arm

It is decided to use a simple robot arm to do the separation. A backup robot arm is also considered.

6.2 Disaster Recovery

I look for many existing measures to decrease the possibility of system failure. The system may fail because of Software, Hardware, and mechanical failure. If we control hazards as much as possible in each part, it is possible to reduce system failure possibility. Having back up plan for each failure is also a good way of avoiding system failure. It must be considered that this system is not safety critical, so back up plans should not elevate cost of system massively.

7 Testing

Both dynamic and static testing have done to verify the system.

7.1 Static testing

ID	Test	Verification
1	Dry run test	Check the program code manually and find possible errors
2	Inspection	Inspect the wiring and input output devices
3	Walkthrough	Check the design with datasheet of components and another expert to get feedback

7.2 Dynamic Testing

Seq#	Test case	Input	Function Under test	Output	Invariant	Assertion	Hazard
1 Dynamic White box	Trash with high velocity	Object	Ultrasonic	Detection through sound waves	Distance of object to the sensor	Echo received by sensor about the object data	H1
2 Dynamic White box	Using fireproof material for the Hull	Flammable trash	Possibility that the machine burning on fire	Sorting the trash	Material shape	No change in material shape	H2
3 Dynamic White box	It's possible to feed the system with trash with mixed-type trash	Mixed-type trash	Camera & Processor	Sort trash in Unknown sub container	Data of mixed-type trash (RGB)	Finding the sample type based on given data set	H3
4 Dynamic White box	Test System in moist environment or with moist trash	Most Trash	The functionality of the system under certain condition	Sorted trash	Moisture level	No damage to electronics and proper sorting	H4
5 Dynamic White box	Input power	Flow of electricity	Processor	Voltage	Voltage between 210v-220v	Perform voltage.	H7
6 Dynamic White box	Battery or input power	Flow of electricity	ATMEL 2560	Voltage	Voltage between 220V	Process data.	H7

7.3 Test cases

Hazard #3

Seq#	Test case Description	Input	Test function	Output	Invariant	Assertion
1 Dynamic White box	It's possible to feed the Sensor with a lot of trash of same type very fast in order to find out the threshold of scanning.	A lot of Trash of same type in sequence	Scanning trash speed	Failing to send scanned data	Trash	Successfully sending the scanned data of trash to the processor.
2 Dynamic White box	It's possible to feed the processor with data of the trash very fast in order to find out the threshold of processing.	Heavy amount of data which needs processing	Processing data speed	Fail to send proper order to the mechanical arm	Data of scanned trash	Proper order to mechanical arm
3 Dynamic White box	It's possible to feed the Sensor with a lot of trash with different types very fast in order to find out the threshold of scanning.	A lot of Trash of different types in sequence	Scanning different trash types speed	Fail to send the scanned data	Trash	Successfully sending the scanned data of trash to the processor.
4 Dynamic White box	It's possible to feed the processor with data of the trash very fast in order to find out the threshold of processing.	Heavy amount of various data which needs processing	Processing various data speed	Fail to send proper order to the mechanical arm	Data of scanned trash	Proper order to the mechanical arm

For Hazard #2:

Seq#	Test case Description	Input	Test function	Output	Invariant	Assertion
1 Dynamic Blackbox	Using fireproof material for the Hull	Flammable trash	Possibility that the machine burning on fire	Sorting the trash	Burning trash	Sorting the trash while there is no fire

7.3 Verification and Validation

First and foremost, the requirements must be confirmed by stakeholders and users. The system has different milestones, after reaching to each milestone the status of improvement should be confirm by stakeholders. Analysis, demonstration, test, and inspection has used for verification and validation.

8 Measures to manage the quality

The system should have regular maintenance.

8.1 Preventive Maintenance

The system will be check in detail by a technician regularly. This checking is consisting of performance of the system, and then performance of subsystems and components.

8.2 Corrective maintenance

If any hazardous event happens, the checking will be immediate. This correction is including repairing and replacing damaged components.

8.3 Suggestion and complains

In case of any complain from users, system should be fixed based on user's requirement. If users have any suggestion which is feasible, it is possible to implement their suggestions after deployment or during development.

8.4 Quality management

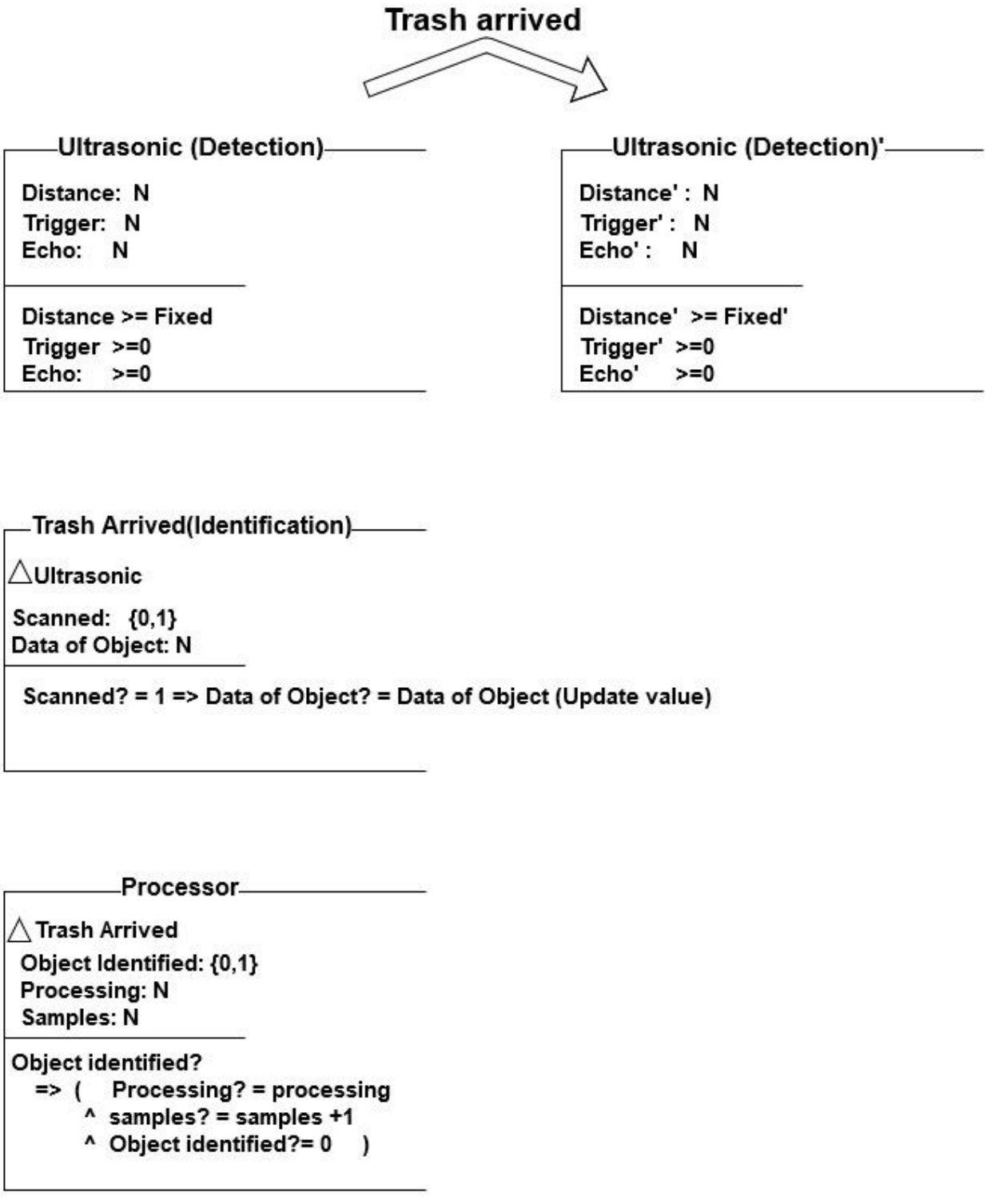
The system follows the newest standard of systems. 9001:2015. Is newest standard that the system follows. This standard insists on continuous improvement of systems.

8.5 Statistical control

Every component has an estimated average life-cycle that is mentioned in its data sheet. Take this matter into account also boost the quality of final system.

9 Formal Method

Formal method is explanation of a system through mathematical basis. Z notation is a useful way of describing behavior of software based on current understanding of mathematics.



10 References

[1] T. W. Staff, "The recycling crisis," *The week*, September 19, 2019 [Online]. Available: https://theweek.com/articles/831864/recycling-crisis.

[2] *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems*, International Standard IEC 61508, 1999.

[3] MacKinnon, L, Raja, K (2019)*: Safety critical Systems*, lecture notes, University of South-Eastern Norway, delivered 14-19 Oct 2019.

[4] A Survey of Approaches Combining Safety and Security for Industrial Control Systems - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Overall-safety-lifecycle-of-IEC-61508-39_fig6_265704308 December 2, 2019 [Online].