By

SOFTWARE MODULE AND TASK SPECIFICATION REPORT

Version 1.0

TEAM 7

Görkem Çakırhan	(HW)
Ceren Canpolat	(SW)
Alper Günçal	(HW)
M.Ali Yıldırım	(MAN)
Fatih Şahin	(IE)

Person Responsible from This Report: Ceren Canpolat
Due Date: December 16, 11

Table of Contents

1 Introduction

1.1 About This Report and to Its Readers

The aim of this report is dividing the project "LessBag" into subsystems in order to achieve desired requirements. Decomposing system into subparts in the designing stage makes the system easily understood and the possible mistakes, which can be made in one of the subparts, can be easily prevented in the implementing stage. This report includes system overview to help reader to see the general working principle, software modules, which are the main tasks to do, and their descriptions to be able to understand the subparts of the system, hardware interface to see the connection between software and hardware part of the system, and finally software system requirements to see if needs of software can fulfill the user requirements which the company MoreMoney explained in the System Requirement Specification Document.

2 System Overview

LessBag is an innovative product that is designed for markets to reduce their consumption on plastic bags. It's a special kind of bag dispenser, which will be mounted near each cash register in the market. Project consists of hardware and software parts:

To be able to understand software part of the system, one should know the general working principles of the cash registers at the supermarkets. In supermarkets there is a main computer that has all the records of sales has been done at the cash registers of the market. After each product has been read by cash register there is a file at the main computer which records the barcode information immediately. The software will read the barcode of the product from this file and from the database that user filled by our database interface it will take the type, mass and volume information for that specific barcode number to decide whether a new bag should be rolled from dispenser. After decision has been made, if a customer needs a bag, software sends a signal containing the id of bag dispenser that should

roll the bag down from main computer's serial port. Software of the system runs continuously by checking the size of "sale file" to see whether algorithm should work or not.

The hardware part has 2 components: control point and bag dispenser. The control point is connected to both the main computer of the market and the bag dispensers to manage the communication between main computer and cash registers. The control point takes the signal from main computer's serial port and processes to see which bag dispenser need to roll a bag down. It sends a signal to that bag dispenser and the control point is done.

The other component of the hardware, bag dispenser is responsible for rolling the bag. There are two actions that cause bag dispenser to roll down a bag: signal may come from control point, or the cashier may press the give-1-bag button. When one of those happens, the bag dispenser rolls down a bag and it's done. After one bag is rolled down, the customer takes the bag himself. The roll won't move, so that the customer will easily pick the bag.

To decide the subparts of the system one should look the main tasks to be done in order to achieve the desired requirements. Therefore, in the system of the LessBag there are five main tasks to be done such as reading the sale file, retrieving product information from database, algorithm which decides whether a bag should be rolled down or not, serial communication between main computer and control point and wired communication between control point and dispenser.

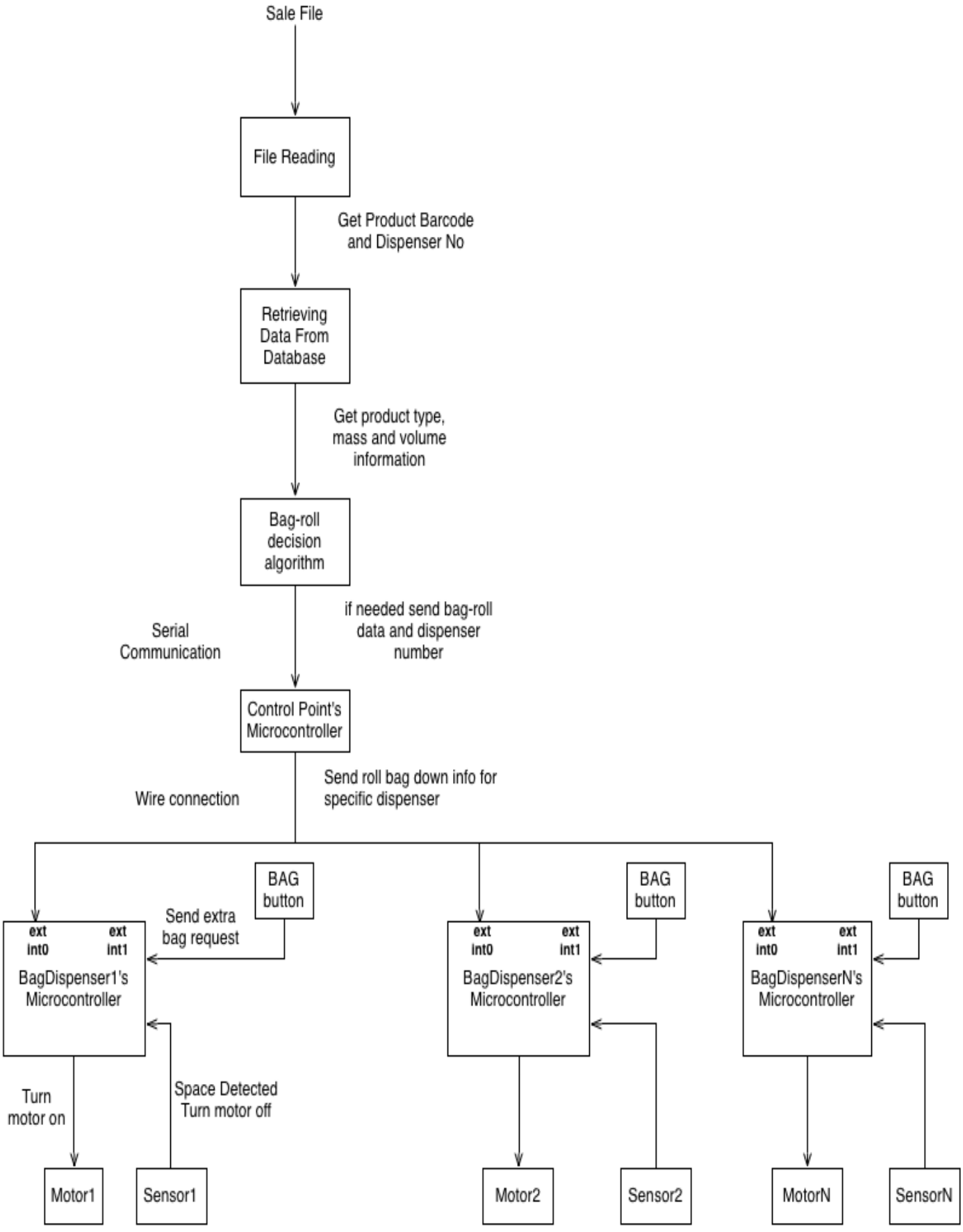

Figure 1:Software System Overview

3 Software Modules

The identification of the main tasks to do which has been mentioned in the system overview will be explained here in details.

3.1 Task 1

The first task to be done is reading the sale file.

- *Classification:* The kind of this component is function.

- *Definition:* Purpose of this task is learning which products the customer has bought.

- *Responsibilities:* Since the sale file has the barcode numbers of products what this component accomplishes is simply reading barcode numbers of the products has been bought from this file. Due to all other components' dependencies to barcode number, this task provides crucial information to rest of the system.

- *Constraints:* When our program interacts with this file it opens and reads all the data inside. However, if the cash register wants to write a new barcode to this file while we were reading it, its content will be changed. Therefore, the constraint is; while reaching to this file, the duration between each product that has been read by cash register should not be too small so that we can read its whole content before it has been changed again. Since, we cannot control this duration, our constraint becomes: the time passes from the start to end of file reading process should be faster than this duration.

- *Uses/Interactions:* This task is being used by Task 2 which is retrieving product information from database. The barcode number file has taken from file is being used to find the type, volume and mass information for that specific barcode number in the database. This task only uses the sale file. If

the file cannot be open for some reason all other components of the system will not be working.

- *Resources:* The resource this task uses is the sale file. This file should not been corrupted by any other programs due to its influence on the overall system of markets such as printing the slip for sold products. Therefore, our program should not interfere the job of cash register in any step of the program and priority of cash register to reach the file should be higher than our program.

- *Interface/Exports:* This component provides the barcode information of the product and the cash register number which the shopping proceeds.

3.2 Task 2

The second task to be done is retrieving product information from database.

- *Classification:* The kind of this component is subsystem.

- *Definition:* The purpose of this component is learning the necessary information of a specific product from database.

- *Responsibilities:* The primary responsibility of this component is to match the barcode number it got from Task1 with barcode column of database and get the type (hygiene, fragile, crushable, 1-bag-required, others), volume and mass of information which belongs to that specific barcode number.

- *Constraints:* In order to take whether a bag should be rolled down or not decision as soon as possible, the barcode matching and retrieving product information from database should take small amount of time. Therefore there is a time constraint here.

- *Uses/Interactions:* This component is being used by Task 3 which is the algorithm which decides whether a bag should be rolled down or not. The type, volume and mass information Task 2 gets from database is being used by

the smart algorithm to be able to fill the correct type of bags with the correct mass and volume information. This component uses Task1 which is reading the sale file in order to get the barcode number of the product has been bought.

- *Resources:* The resource of this component is database. Since there is only one access to database, and it is from main computer, there is no possible race condition or deadlock.

- *Interface/Exports:* The data provided by this component is the type, volume and mass information of the product. The meaning of "Type" value is the product's type which can be hygiene, fragile, crushable, 1-bag-required or anything else (others). The meaning of "Volume" value is the volume of the product in terms of centimeter cube. The meaning of "Mass" value is the mass of the product in terms of gram.

3.3 Task 3

The third task is the algorithm which decides whether a bag should be rolled down or not.

- *Classification:* The kind of this component is function.

- *Definition:* The purpose of this component is to determine whether a bag should be rolled down from dispenser after the addition of the new product which has been read by cash register.

- *Responsibilities:* The main responsibility of this component is deciding whether a bag should be rolled down from dispenser as it has been mentioned in the definition. The way to do this is; hypothetically filling the specific open bags, which are the ones already rolled down from dispenser, by the help of all necessary product information (type, volume and mass). If bags for a type is full then for the new product in that type a new bag should be rolled down.

- *Constraints:* There is a limitation for time in this task. Since time for a shop is considerably small if the cashier is fast than the deciding whether a new bag should be rolled down or not should be fast too. Therefore algorithm should be as efficient as possible to reduce the time it requires to provide the desired output. Also since this component runs in an infinitely loop the code should be small not to disturb the main component.

- *Uses/Interactions:* This component uses Task2 which is retrieving product information from database, and used by Task4 which is serial communication between main computer and control point. The product information we get from database is used as input in Task3; and the output of this task, which is the input of Task4, is a variable which states the necessity of bag-rolling operation.

- *Resources:* There are no resources that are managed, affected or needed by this entity except the product information which has been given by Task2. Since this information is not external to design it cannot be counted as a resource.

- *Interface/Exports:* The data has been provided by this component are the decision variable of bag-roll operation and the variable which tells the number of cash register who need the bag. The decision variable is 1 if a bag should be rolled down, 0 if no bag necessary yet. The cash register variable is simply an integer which holds the cash register number.

3.4 Task 4

The fourth task is serial communication between main computer and control point.

- *Classification:* The kind of this component is subsystem.

- *Definition:* The specific purpose of this component is to deliver the bag-roll request to control point if needed.

- *Responsibilities:* The primary responsibility of this task is to take bag-roll request from Task4 and send it to microcontroller of the control point serially.

- *Constraints:* Since when it comes to communication between two devices there is a chance that a data loss can be occur, the data come with serial communication should be checked at the destination in order to confirm that no error occurred. Therefore a checksum can be added at the source to the data and can be checked if it is correct at the destination. Since time is gold to LessBag, the duration of communication should last small. In order to include fast shopping possibility this serial data should be stored at control point.

- *Uses/Interactions:* This component uses Task3, which is the algorithm decides whether a bag should be rolled down, in order to know if a bag-request exists. Task5 uses this component in order to know if a bag request exists and which cash register's bag dispenser needs a bag-roll operation.

- *Resources:* The only resource which has been affected by this task is the memory in microcontroller in order to save the data come from serial port. There is no any possible race conditions or deadlock situations here.

- *Interface/Exports:* The data provided by this task is a frame of data which contains the bag roll request, the specific cash register number for that request and a checksum.

3.5 Task 5

The fifth task is wired communication between control point and bag dispenser.

- *Classification:* The kind of this component is subsystem.

- *Definition:* This task's aim is to deliver the bag roll request from cash register to the correct bag dispenser.

- *Responsibilities:* This task responsible from taking the data from control point, process it to understand which bag dispenser is required to roll a bag, and send a signal to that correct bag dispenser.

- *Constraints:* Since the control point will send a signal to the bag dispenser, the dispenser's microcontroller should always wait for a signal that may come from control point's microcontroller. Also the size of the signal should be long enough so that the dispenser can catch it.

- *Uses/Interactions:* This task uses Task4 which the serial communication between main computer and control point. Since the data which contains the bag-roll request comes from Task4, Task5 cannot operate without Task4.

- *Resources:* Since the microcontroller of the bag dispenser should wait a possible signal that may come from control point, the microcontroller's whole processing capability is reserved for this. This problem can be solved by using external interrupt. If external interrupt would be used for this purpose, there would be no sources allocated for this task.

- *Interface/Exports:* The signal that this task sends is just a 0 to 1 transition, to let the bag dispenser know that it should roll down a bag.

4 Hardware Interface

The hardware components that are used in LessBag are:

- Microcontrollers: Control point's microcontroller and bag dispensers' microcontrollers
- Motors: Bag dispensers' motors
- Sensors: Bag dispensers' sensors
- BAG buttons

Main computer directly controls microcontroller of the control point. The input of this microcontroller is the output of Task4 which is the bag-roll request and number of bag dispenser, which is the owner of the request, from serial connection. This connection established from serial port of main computer to RxD pin of microcontroller.

Microcontrollers of the bag dispensers are being controlled by control point's microcontroller. There are wires which has been mounted from pins of the control point's microcontroller to each of the bag dispenser's microcontroller's external interrupt 0 pin. This connection is established by Task5 which is sending a 0 to 1 transition signal from control point's microcontroller when a bag-roll request exists. When bag dispenser's microcontroller gets this interrupt signal, it automatically goes to roll-a-bag-down process state.

In this state bag dispenser's microcontroller turns on the motor and starts to roll down the bag. In order to understand whether one full bag has been rolled down or not the sensor sends an infrared light to bag and when the space, which is the beginning of the bag, occurs the sensor will send turn off the motor data to one of the port of bag dispenser's microcontroller.

The BAG button is connected to ex int1 pin of the bag dispenser's microcontroller. Whenever someone presses this button the microcontroller will automatically go to roll-a-bag-down process state.

5 Software System Requirements

Time Requirements: Since LessBag need to be fast in order not to wait the customer time requirements for each task has crucial importance. Time requirements for each task is:

- Task1: Reading the file in order to get the barcode number of product should not take more than 50 milliseconds since reading the file can cope with this time constraint this task can accomplish its time requirement in time.
- Task2: Retrieving the product information from database should not take more than 100 milliseconds in order to make the bag-roll decision faster. This is easy to achieve if main computer's hard disk's seek time and rotational latency are low enough and its ram should have enough space.
- Task3: Algorithm which will make the bag-roll decision should not take more than 50 milliseconds since the code for it is not that long this decision duration will achieve this time constraint.
- Task4: Serial communication should not exceed 10 milliseconds in order to satisfy the general system's time constraint. Since the baud rate is the number of times the signal can switch states in one second, if we operate at 9600 baud, the line can switch states 9600 times per second. This means that each bit has the duration of 1/9600 of a second. This corresponds to 100 microseconds. Since we will send at most 40 bits, the duration will be at most 4 milliseconds.
- Task5: Sending a 1 to 0 transition signal over a wire should not take too much time. Therefore in 10 milliseconds this task can be and should be completed.

Space Requirements:

- Task1: The space for task one is simply the size of sale file, this file's size depends on the shopping has been made therefore it cannot be interfered. However if we assume that there can be at most 30 products in one shop the size of the file can be at most 18KB.

- Taks2: The space required for database should be large since there are thousands of products recorded inside it. The total space it should use should be at most 100KB for 1000 products.
- Task3: The space for the decision algorithm should be small because the algorithm has not uses too much space already. However we can say that it should not exceed 5KB.
- Task4: Only space this task can use is the memory of microcontroller of the control point. Since we will store the serial data in order to coping with very fast rolling down requests, the microcontroller which has memory of at least 2K will be enough for us.
- Task5: Since this task only sends an interrupt signal it has no space requirements.

Hardware Requirements: All the hardware components described above satisfies the software needs of each task since microcontrollers have been chosen according to these needs.

6 References

1. “Serial Communication”.
   <http://www.wcscnet.com/Tutorials/SerialComm/Page1.htm>
2. “System Context Diagram”.
   <http://en.wikipedia.org/wiki/System_Context_Diagram#Building_blocks>
3. “Database Constraints”.
   <http://en.wikipedia.org/wiki/Category:Database_constraints>
4. “Database Management System”
   <http://en.wikipedia.org/wiki/Database_management_system>
5. ”System Requirements”
   <http://en.wikipedia.org/wiki/System_requirements>