LESSBAG

by

MOREMONEY

PRELIMINARY DESIGN DOCUMENT

TEAM 7

Görkem Çakırhan
Ceren Canpolat
Alper Günçal
Muhammet Ali Yıldırım
Fatih Şahin

08.12.2011

Table of Contents

1 INTRODUCTION

1.1 About This Document and to Its Readers

Preliminary Design Document describes the product in terms of internal structures and how those structures are related with each other. It includes a block diagram to help the reader to understand the hardware part, and a flowchart to understand the software part easily. The reader can understand how the product works without knowing technical details.

1.2 Product 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:

In the software part, a smart algorithm is implemented to calculate required number of bags by checking the products that the customer bought. The software has a database which stores the barcode numbers of the product and the product's volume, mass and type info. The algorithm checks the volume, mass and type information to calculate the optimal plastic bags required to package those products. Software runs continuously and makes calculations to see if a customer needs a bag or not. After calculating, if a customer needs bag, software sends a signal containing the id of bag dispenser that should be rolled down, from main computer's serial port. Software is done, it continues calculating again.

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 is needed to be rolled 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.

2 HARDWARE

2.1 Block Diagram of the System and Subassemblies

LessBag’s hardware part consists of 2 subassemblies of LessBag: control point and bag dispenser. Figure-1 is the system block diagram which illustrates the general overview of the hardware part of the product.

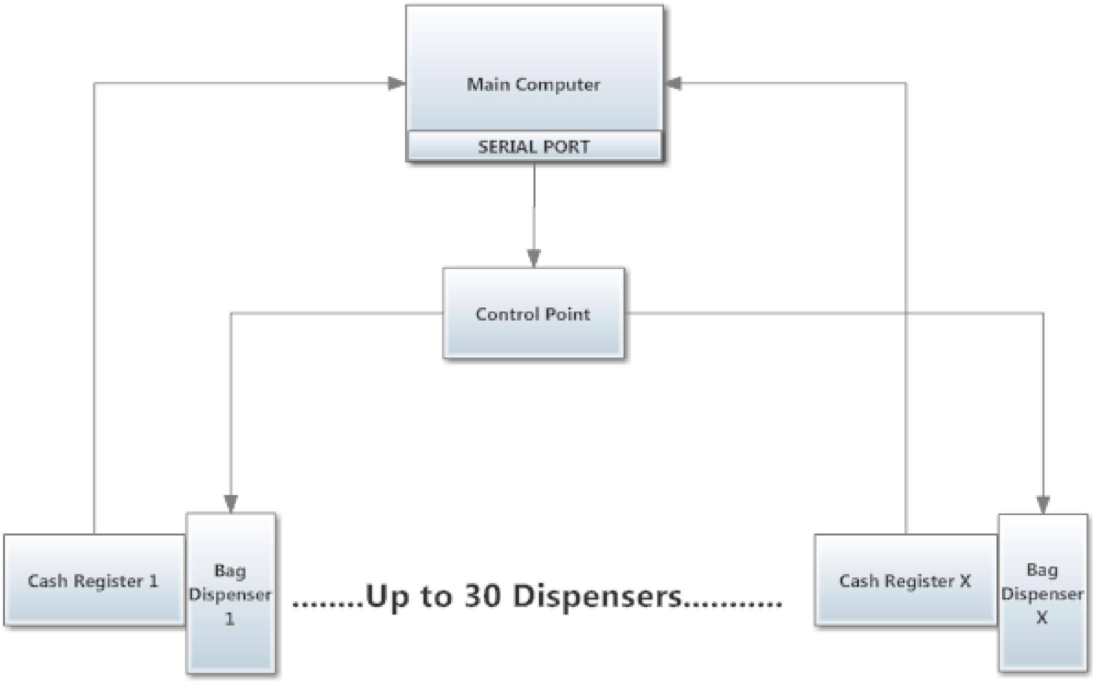


Figure-1: System Block Diagram

After main computer calculates and decides that a new bag should be rolled down from a dispenser, say dispenser number X, it send data to control point from its serial port. That data actually says to control point that: “Hey control point, say bag dispenser number X that it should roll down a bag.”

2.1.1 Control Point

After control point processes the signal and understand that command, it send a signal to the bag dispenser number X's port. To understand how control point works, a closer look to the control point is shown in Figure-2.

Control point takes the information from main computer through serial port, and sends the roll down request to related dispenser. Figure-2 illustrates a closer look to control point.

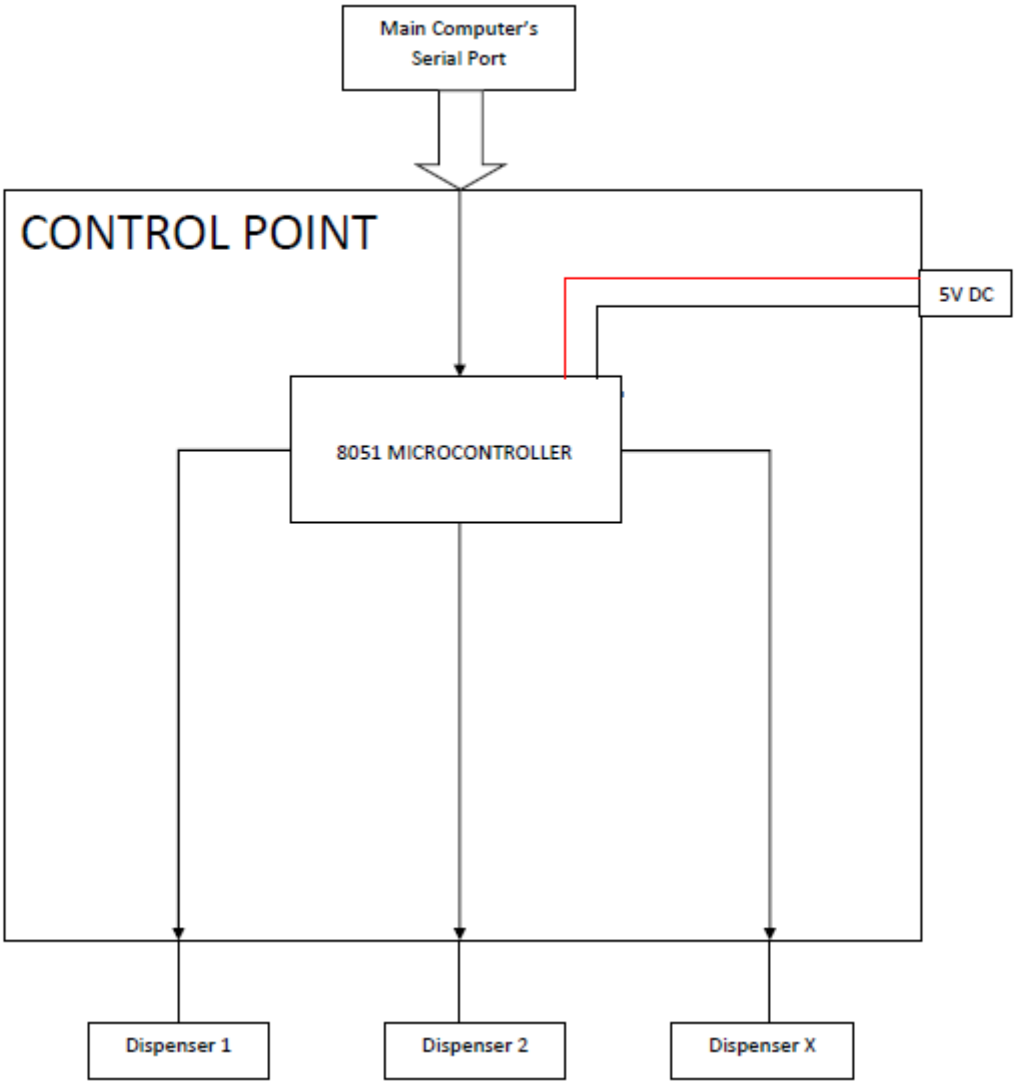


Figure-2: A closer look to the one of subassemblies of LessBag: Control Point

As can be seen from the Figure-2, control point only consists of a microcontroller. Although this device seems to be a very simple device, it has a very crucial role. Without this device, there would be only one lucky cash register that would have bag dispenser near it: Main computer would be only connected to one bag dispenser. Control point let many bag dispensers to connect to main computer.

2.1.2 Bag Dispenser

The other subassembly of LessBag is the bag dispenser. After control point receives the command from main computer which was telling bag dispenser X to give one bag, it sends a signal to bag dispenser X. Dispenser X is always waiting signal from control point, and when that signal comes, it means “Hey dispenser X, you should roll down one bag, but exactly one bag, no more, no less.” After dispenser X receives that signal, it immediately switches on the relay to start the DC motor. There is an infrared sensor in the bag dispenser to detect whether one bag is rolled down from the dispenser. If one bag is rolled down, sensor tells the microcontroller that, “Hey microcontroller, that’s enough, one bag is rolled down”. Then microcontroller stops the engine by switching off the relay.
There is another probability that bag dispenser runs that same bag giving process: the cashier presses the give-1-bag button to give extra one bag to the customer if required. In case that button is pressed, the dispenser does the same bag giving process again. For better understanding, a closer look to the bag dispenser is shown in Figure-3.

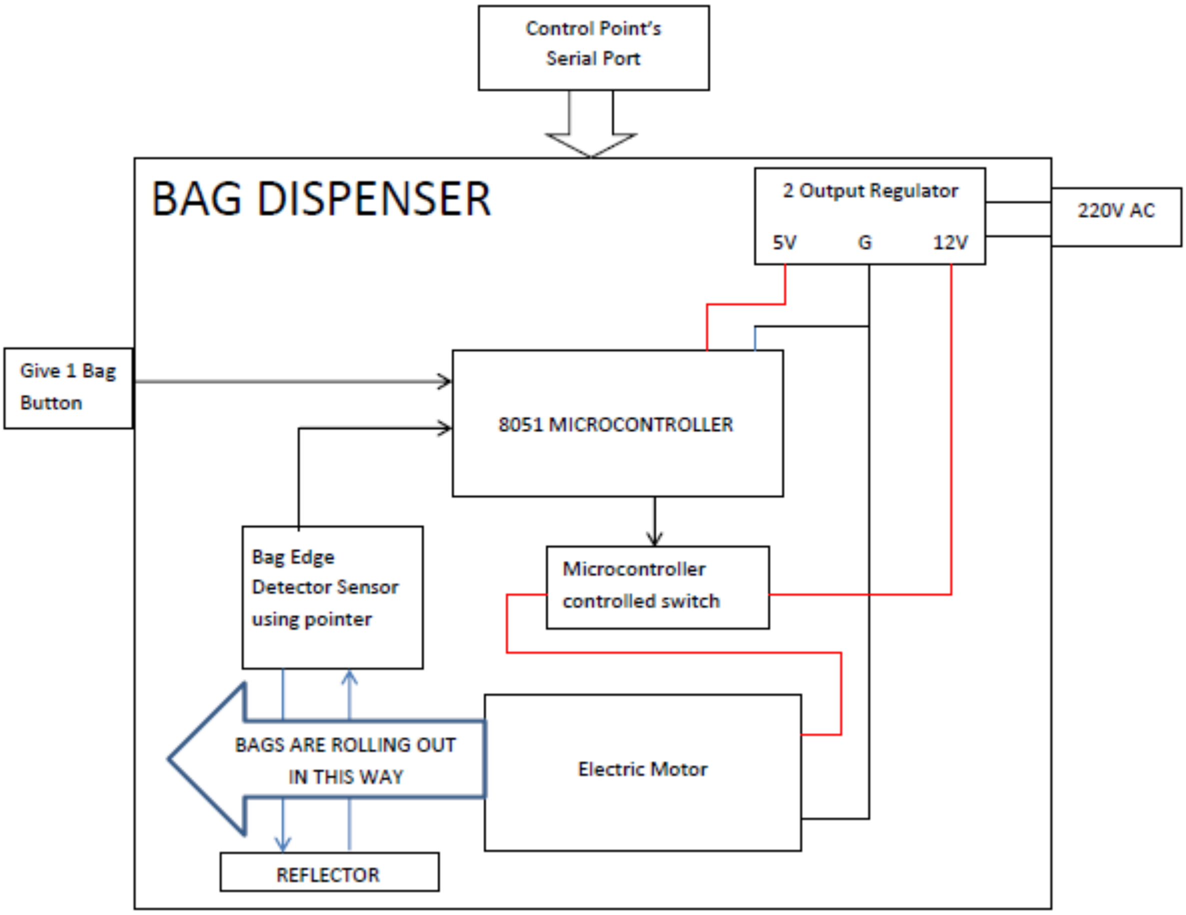

Control Point’s Serial Port
BAG DISPENSER
2 Output Regulator
5V
G
12V
220V AC
Give 1 Bag Button
8051 MICROCONTROLLER
Bag Edge Detector Sensor using pointer
Microcontroller controlled switch
BAGS ARE ROLLING OUT IN THIS WAY
Electric Motor
REFLECTOR

Figure-3: A closer look to the one of subassemblies of LessBag: Bag Dispenser

2.2 Product Tree and BOM

Product tree telling the same thing as block diagram, except it is more like top view. It excludes some details but describes the system as a whole. Figure-4 is the product tree of the LessBag.

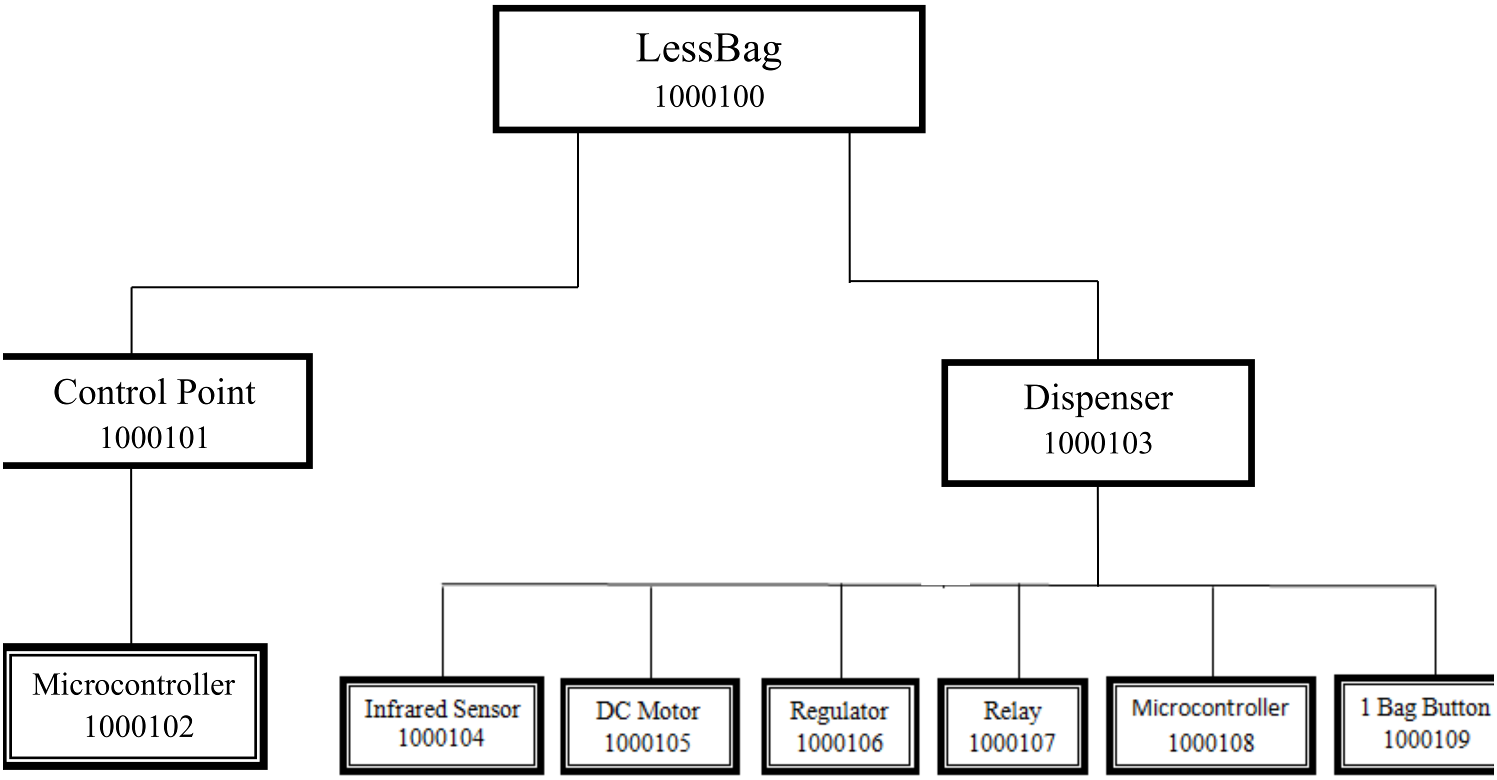


Figure-4: Product Tree of the LessBag

The Bill Of Material (BOM) is shown in Figure-5. BOM is a table indicating the main materials in the Product Tree. It contains stock number, component description and a number showing the number of components of the given row used in the assembly.

Assembly Stock Number	**Assembly Part Description**	
1000100	LessBag	
Stock Number	**Part Description**	**Number Used**
1000101	Control Point	1
1000103	Dispenser	1
Drawing no	Prepared by	Checked by
1000100-BOM	Alper Günçal	Görkem Çakırhan
Assembly Part Description	**Date**	
LessBag	08.12.2011	

Figure-5: Bill Of Materials (BOM)

3 SOFTWARE

3.1 Smart Algorithm

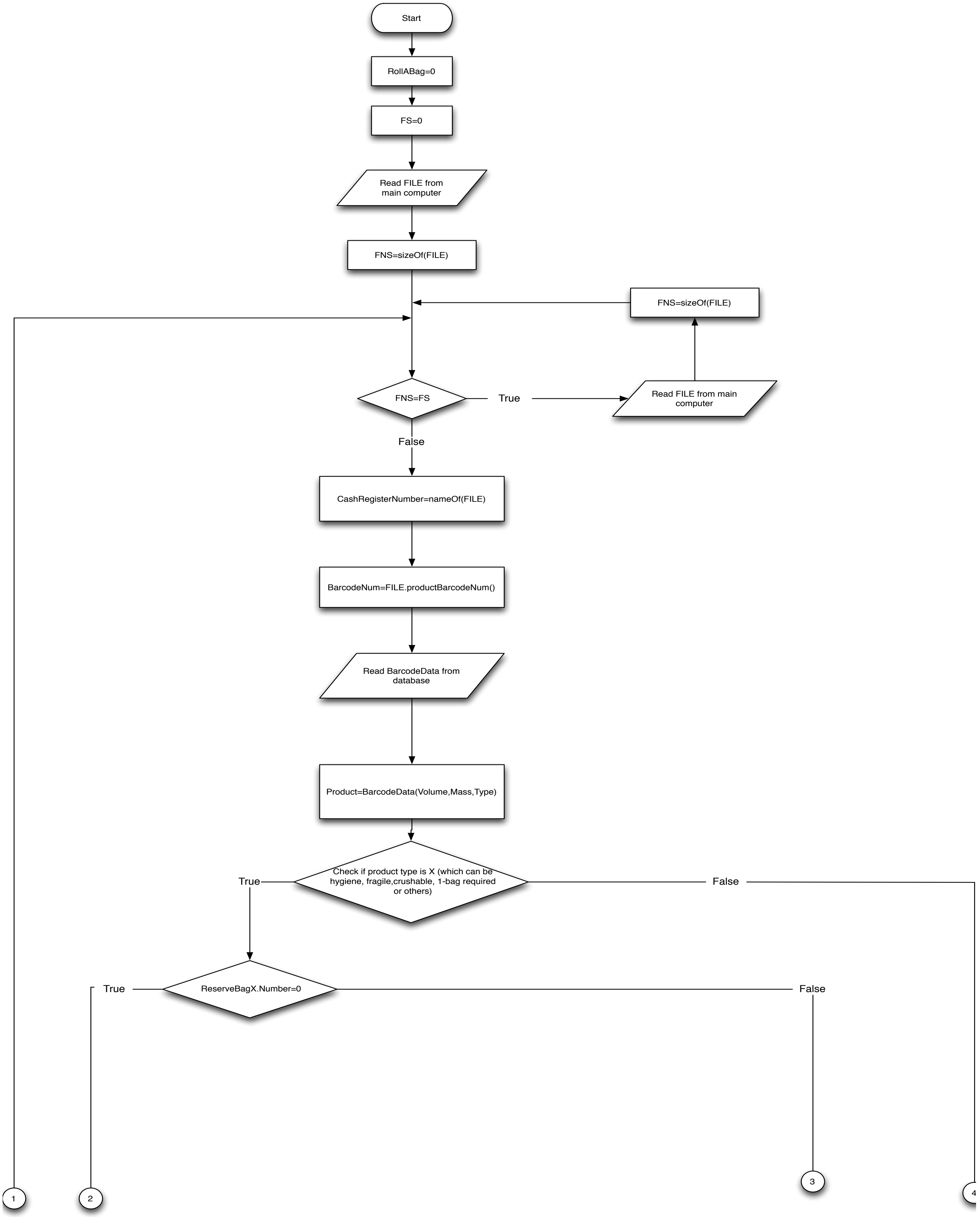
The system, whose flowchart is below, uses an algorithm in its software and hardware part to perform its functions. Our algorithm for this system is as follows:

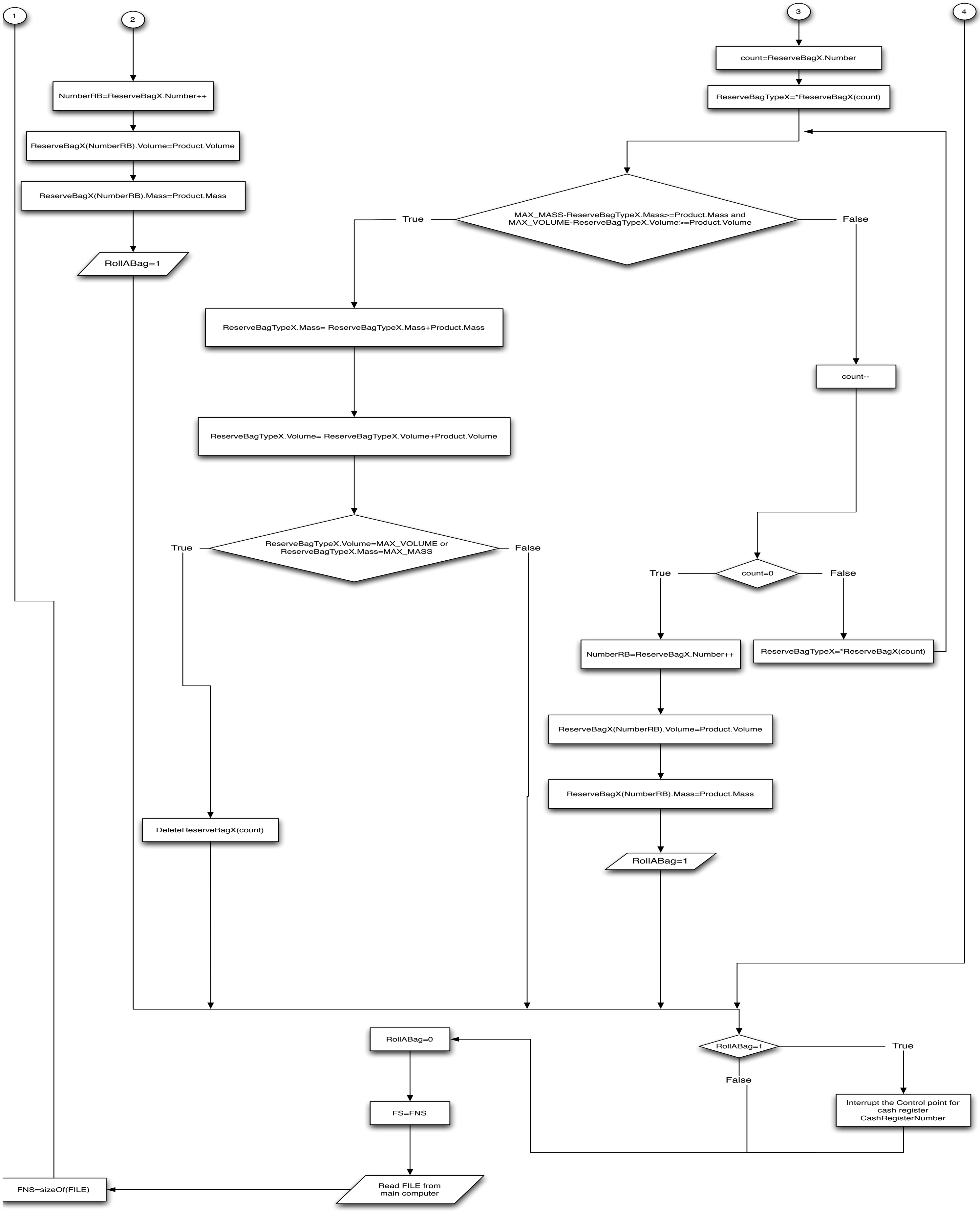
Our program monitors continuously each cash register's specific file whose size changes when there is a selling process. After understanding that shopping starts at a specific cash register, say cash register number A, we take the barcode number of the sold product from the specific file of cash register A. We find the product that corresponds to that barcode number in our database, and take the volume, mass and type information. In our database we have 5 different types which products belong to: Hygiene, crushable, breakable, 1-beg-required and others. Those different categories imply that the products which are from different categories needed to packed into different bags with the other products in their own categories. Note that each bag has a maximum volume and mass capacity and those maximums should not be exceeded. And all reserved bags have a remaining volume and mass capacity.

According to the type information of the product say X we categorize it, and then we check whether there is a reserved bag for that category or not. If there is not any reserved bag for that specific category, we reserve a bag for this new category. If there are already reserved bags, we check each reserved bag for this specific category to see if there is enough remaining volume and mass for this new product inside. If there is enough space in a bag, we put this product into that bag and update the remaining volume and mass information of that bag. If there is not enough space in any reserved bag, we reserve a new bag. Once a bag is reserved, it will directly roll-down from our dispenser. This process will continue for each product till the shopping ends. This algorithm is for calculating the required number of bag.

3.2 Database Interface

There will be an interface available to user which provides CRUD operations. These operations which are create read update delete will help user to enter all necessary information about product in case there exists a new product or delete all the information of a product which is no longer available or update all the information such as new weight or volume of product or read a specific information of the product.


Start
RollABag=0
FS=0
Read FILE from main computer
FNS=sizeOf(FILE)
FNS=sizeOf(FILE)
FNS=FS
True
Read FILE from main computer
False
CashRegisterNumber=nameOf(FILE)
BarcodeNum=FILE.productBarcodeNum()
Read BarcodeData from database
Product=BarcodeData(Volume,Mass,Type)
True
Check if product type is X (which can be hygiene, fragile,crushable, 1-bag required or others)
False
True
ReserveBagX.Number=0
False
1
2
3
4


1
2
3
4
count=ReserveBagX.Number
ReserveBagTypeX=*ReserveBagX(count)
NumberRB=ReserveBagX.Number++
ReserveBagX(NumberRB).Volume=Product.Volume
ReserveBagX(NumberRB).Mass=Product.Mass
RollABag=1
True
MAX_MASS-ReserveBagTypeX.Mass>=Product.Mass and MAX_VOLUME-ReserveBagTypeX.Volume>=Product.Volume
False
ReserveBagTypeX.Mass= ReserveBagTypeX.Mass+Product.Mass
count--
ReserveBagTypeX.Volume= ReserveBagTypeX.Volume+Product.Volume
True
ReserveBagTypeX.Volume=MAX_VOLUME or ReserveBagTypeX.Mass=MAX_MASS
False
True
count=0
False
NumberRB=ReserveBagX.Number++
ReserveBagTypeX=*ReserveBagX(count)
ReserveBagX(NumberRB).Volume=Product.Volume
DeleteReserveBagX(count)
ReserveBagX(NumberRB).Mass=Product.Mass
RollABag=1
RollABag=0
RollABag=1
True
False
FS=FNS
Interrupt the Control point for cash register CashRegisterNumber
FNS=sizeOf(FILE)
Read FILE from main computer

4 REFERENCES

1) “ticaretimiz." N.p., n.d. Web. 17 Oct 2011. <http://ticaretimiz.com/p84045-photocellifotoselli-emotion-kagit-havlu-makinasi.html>.

2) “8951 8 bit Controller”. <http://www.datasheetcatalog.org/datasheets/134/411652_DS.pdf>

3) “MySql”. http://www.mysql.com/

4) Absolute RLS125 Relay.< http://www.sonicelectronix.com/item_31162_Absolute-RLS125.html>

5) “INFRARED SENSOR”. <http://www.pololu.com/catalog/product/958/>