Supermarket Checkout - 1

More Soft Systems
for Logicator

ACTIVITIES

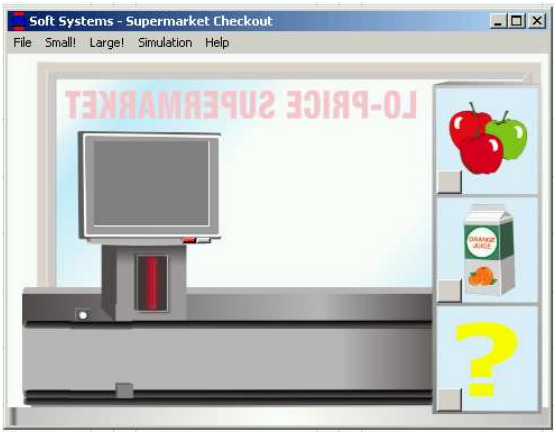
These materials present a series of activities for building programs to control the Supermarket Checkout Soft System. Together, they build into a complete control system for the supermarket checkout.

Skills - **You need to know how to:**

- Switch outputs (including motors) on and off in a timed sequence;
- Monitor both digital and analogue input devices and produce an output in response;
- Use procedures as building blocks to build a larger program.

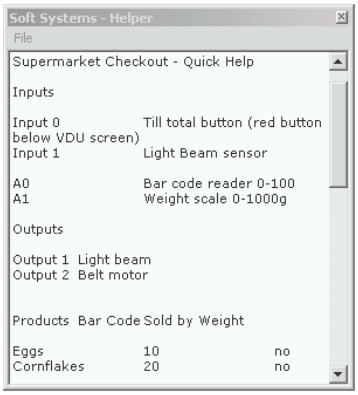
These skills are taught in the teaching materials for the: Pelican Crossing, Car Park, Burglar Alarm, and Greenhouse.

From the **System> Soft Systems** menu, select **Supermarket Checkout**



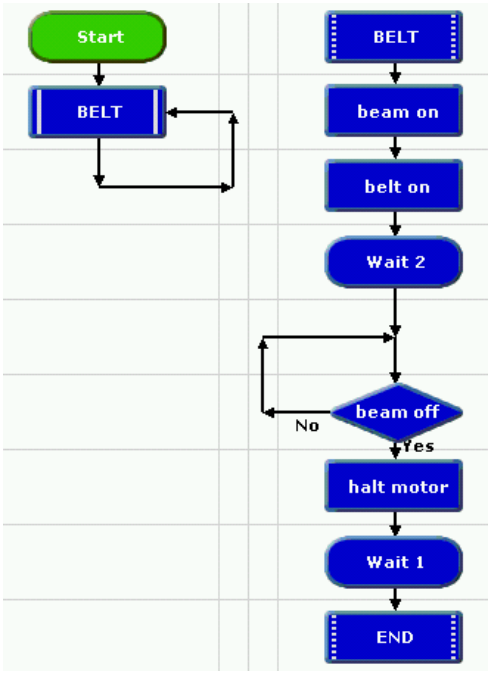
From the Soft System window menu, select **Help>Quick Help**

Look through this carefully to see what input and output devices there are in this system.

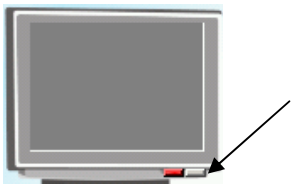


Activity One – Controlling the belt

Build the program shown below:



Now follow these instructions to investigate how the checkout works:



1. Click the grey button under the screen to switch on the display.

2. Run the program.

3. Click the "random non-weighed items" button.

When this item reaches the checkout, the screen displays its bar code number and name.

4. Click the "random weighed items" button.

When this item reaches the checkout, the screen displays its bar code number, name and weight.

Supermarket Checkout - 2

More Soft Systems
for Logicator

Activity Two – Creating a bill

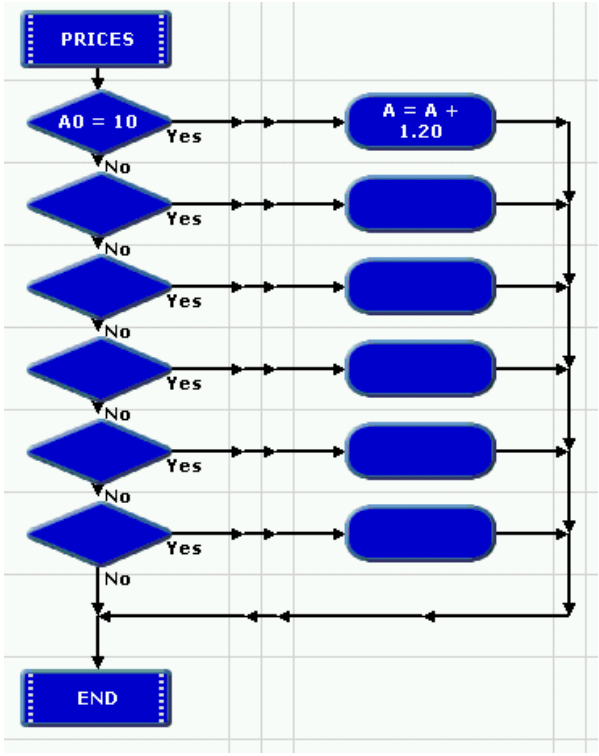
The six different non-weighed items that appear on the belt are listed in the table below. Each one has a "bar code" which is read by sensor A0.

1. The table includes a price for eggs. Set your own price for each of the other non-weighed items.

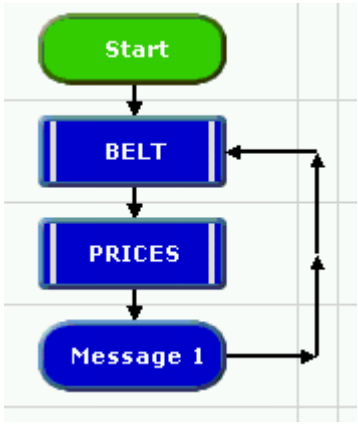
Item	Bar code (A0)	Price (£)
Eggs	10	1.20
Cornflakes	20	
Beans	30	
Bread	40	
Juice	50	
Milk	60	

2. Add a procedure called PRICES to your program, as shown below. This uses Compare commands to check the "bar code" number of the item (A0). Each Compare command is followed by an Expression command that adds the price of that item to the counter A.

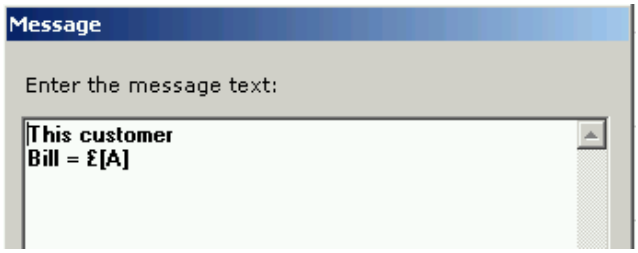
The first pair of commands has been completed for you. Complete the rest for yourself.



3. Add a Do Proc. PRICES command to the main routine in the program. Then add a Message command, as shown below.

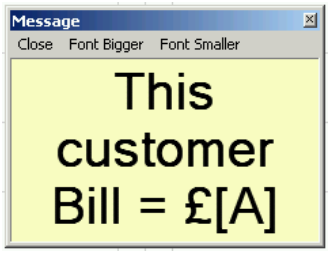


4. Double click on the Message command to open it. This message will display the running total of the bill, so type in the following text:



Make sure that you include A in square brackets [A] so that when the program runs, the Message will display the current value of A.

When you enter the text, the Message will appear on screen. You can drag the corner of the box to vary its shape and size. Click "Font Bigger" or "Font Smaller" to set the size of text you want.



5. Run the program. Click the "random non-weighed items" button. The Message should display the running total for the bill.

With this system, you have to stop the program to start a new bill. The next section shows how you can display a total for each customer, and then carry on to create a bill for the next customer without stopping the program.

Supermarket Checkout - 3

More Soft Systems
for Logicator

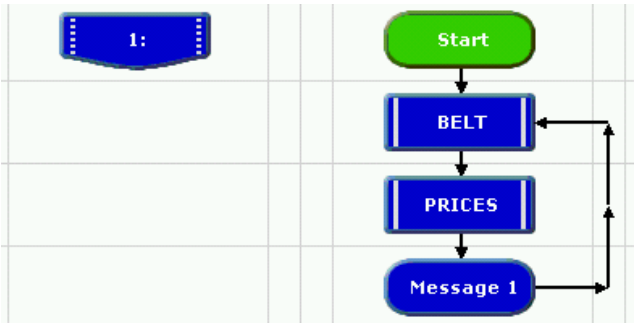
Activity Three – Creating a total for each customer

The red button under the screen is used to signal a change of customer.

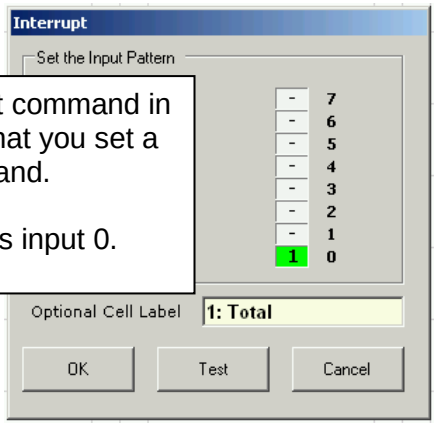
To make use of this button, you need to use Logicator's **Interrupt** feature. An Interrupt is a special kind of Procedure that is called up by pressing a switch in the system. Whenever you press the switch while the program is runnning, it will respond instantly.

Follow these instructions to add this feature into your Supermarket Checkout system.

1. Place an Interrupt command beside the main routine.



Double click on the Interrupt command.

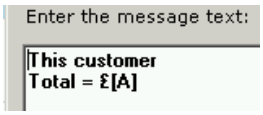


Set the Interrupt command in the same way that you set a Decision command.

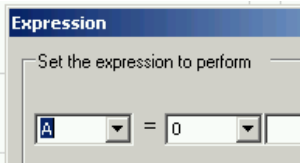
The red button is input 0.

2. Add commands under the Interrupt command as shown below. Use the notes beside each command to help you.

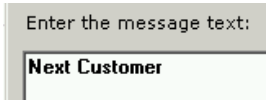
Message 2 text:

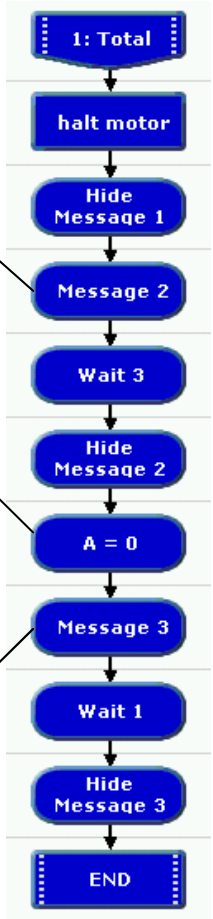


This Expression command resets the counter A back to zero ready for the next customer.



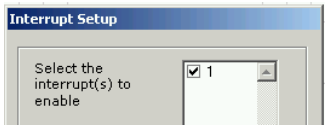
Message 3 text:

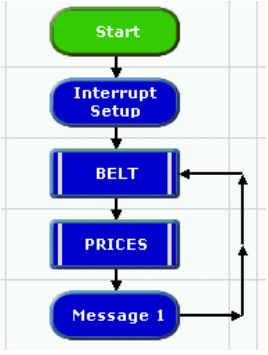




3. The program needs to be set up to react to the Interrupt, so place an Interrupt Setup command under Start as shown below:

Double click on the Interrupt Setup command, and check the Interrupt 1 box:





4. Run the program. When you click the red button, the first customer's total bill should be displayed for three seconds. Then the system should reset ready for the next customer.

Supermarket Checkout - 4

More Soft Systems
for Logicator

Activity Four – Including weighed items in the bill

The four different weighed items that appear on the belt are listed in the table below. Each one has a "bar code" which is read by sensor A0.

Each item is automatically weighed. The weight is read by sensor A1, and displayed in kilos on the checkout screen. Note that every time the item appears it has a different weight.

The table includes a price for carrots. Set your own price per kilo for each of the three other weighed items.

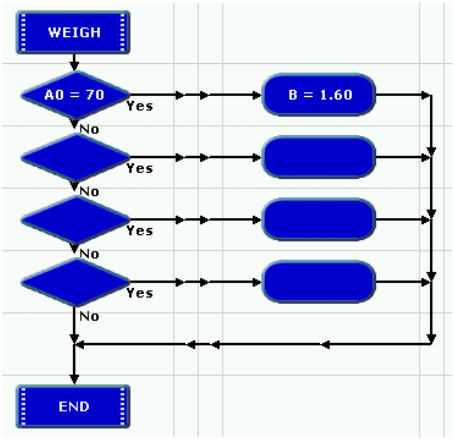
Item	Bar code (A0)	Price per kilo (£)
Carrots	70	1.60
Grapes	80	
Bananas	90	
Apples	100	

Specification

The system must calculate the price of each weighed item by multiplying the weight of the item (the reading from sensor A1) by the price per kilo of that item. It must then add the calculated price to the bill.

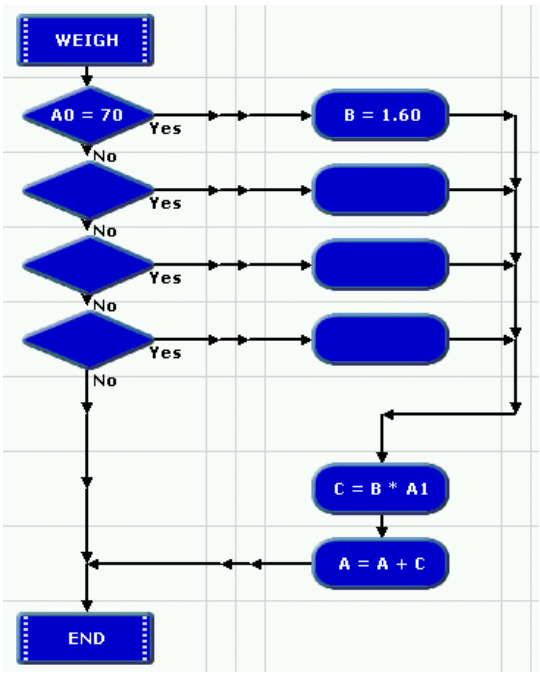
You could try developing this part of the system for yourself, or use the following information to help you to add it to your program.

1. The procedure below uses a counter B, to record the price per kilo for each weighed item:

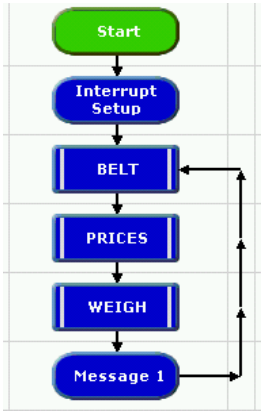


2. If you add two Expression commands to the procedure as shown below, it will also calculate the price of the item and record it in counter C (C = B * A1).

The second Expression command adds this price to the running total of the bill (A = A + C)



3. The main routine must include a Do Proc. command to call the WEIGH procedure.



Run the program to test it. Use the "random product " button to generate weighed or non-weighed items randomly.