Jahan Afshari
Etec 471
Barcode Scanner System
Final Project Description
Dec. 13, 2002
Introduction

I propose a project that will improve the operation of the IEEE snack counter that is unmanned. The operation of this snack counter is characterized by keeping inventory of products sold, counting the revenue from the sales, and keeping track of student tabs.

Keeping inventory requires accurate counting and plenty of time if it is done manually. The expected revenue from the sales is derived from keeping inventory and therefore depends on the accuracy of the person counting how many products are left. Finally, student tabs (recorded on a tab sheet) pose a security issue because a person can add to another person’s tab and that person will probably get away with it.

The project that I propose to use at this snack counter is a barcode scanner system. The user will be able to program a name and a price to a certain barcode number (in UPC code) on a product. Also, the project will keep track of transactions and what was sold at each transaction. Also, the project will have an account system for people that want to keep tabs. People that have accounts will use the barcode on their ID cards to alter their tab balance so that no one can add to another person’s tab. The product information and the accounts will be able to be deleted from the system.

Description/Sketch of Project Hardware and Applications

The barcode scanner system hardware block diagram is shown in figure 1. The system will be a stationary device which will be powered through a wall transformer. There will be three distinct components for the final design of the product: the transformer, the barcode wand, and the rest of the hardware in the main enclosure. Of course, there will be a cable from transformer to the main enclosure and a cable from the barcode wand to the main enclosure. The barcode wand will be used to select the information of a discreet product via the UPC code
Figure 1: Barcode Scanner System Hardware Block Diagram

and allow users to select private information via the barcode on the user's student identification card.

The main enclosure will contain the LCD, keypad, serial EEPROM, reset circuitry, and microcontroller. The LCD will be the tool by which the user will view the selected information and will guide the user in options before and after information has been selected. The keypad will be the tool by which the user will actually respond to these options that are displayed. The options will be discussed in detail in the user interface section of the functional description. The serial EEPROM will store the nonvolatile information: the product price information, the product tallies, student tab balances, total revenue from regular transactions, and the sum of all tab balances.

The barcode scanner system product sketch is shown in figure 2. The barcode scanner system will be designed to read UPC (used on products) and Codabar (used on student ID cards).
Figure 2: Barcode Scanner System Product Sketch
symbologies. There will be two distinct groups of users: students that may or may not have tab accounts and the IEEE officers (administrators). The administrators will be responsible for updating product price information and using the inventory and revenue information; also, they will be in charge of collecting tabs that aren't being paid. Also, each administrator will be able to add or remove administrators from the system.

The maximum dimensions of the main enclosure are 8 in. x 6 in. x 3 in. and the maximum dimensions of the barcode wand are 7 in. x 0.5 in. x 0.5 in.

**Hardware Functional Description**

The functional hardware block diagram is shown in figure 3. The main items of hardware include the LCD, 16 button keypad, barcode wand, serial EEPROM, wall transformer power supply, reset circuitry, and M68HC912B32 microcontroller. Each of these items and their use in the project will be separately described:

- **Barcode wand**

  The wand, which is pen shaped, provides an illumination on the barcode and detects the reflection of light from the barcode to determine if the color at an instantaneous time is black or white. If the color is black, then the digital output will be high; if the color is white, then the digital output will be low.

- **M68HC912B32 and Serial EEPROM**

  The microcontroller will handle the tasks of decoding the output of the barcode wand, responding to the keypad entries, and updating the display on the LCD. The resources that will be used on the microcontroller are shown in figure 3.
The memory usage of the barcode scanner is as follows: the 32kbytes of Flash EEPROM will be used for the program code and some product and price information that will stay constant. The 1kbyte of RAM will be used for the variables of the program code and the stack. The serial EEPROM will store the nonvolatile information that needs to be changed during operation: the product price information, the product tallies, student tab balances, total revenue from regular transactions, and the sum of all tab balances. The EEPROM must be \( \geq 8 \)kbytes and SPI compatible. As shown in figure 3, there are separate read/write data lines for the EEPROM.

*Figure 3: Functional Hardware Block Diagram*
• Reset Circuitry  
   The reset circuitry will serve as a low voltage detector for the microcontroller so that weird operation of the microcontroller will not occur.

• 2x16 LCD and 16 button keypad  
   The LCD will be a 2 row by 16 column screen that displays characters (numbers, letters, and symbols). It will be used to display selected information. The keypad will be a 16 button 4x4 matrix type. The legend shown in figure 2 is the one that matches my project; the original one is arranged similar to a phone keypad with 4 other buttons. Selection between numbers and letters will be similar to that of a cell phone where the button is pressed rapidly to get the character desired and then that character is selected after about a 1 second wait.

• Wall Transformer Power Supply  
   The power supply will be a wall transformer with a 120V ac input and a regulated 5V dc output with an output current of at least 100mA.

Software Description

   The software for the barcode scanner system will be written in the C programming language. The preemptive kernel MicroC/OS-II will be used to handle the various tasks in the software. Code by Logan Haines for using the serial EEPROM with SPI will be adapted for this project. The following is the list of the modules that will be in the software:

   LCD- this is an already existing module that will control the LCD.

   KERNEL- this module will incorporate MicroC/OS-II to handle task switching, scheduling, etc.

   BARCODE- this module will sample the output of the barcode wand at a certain rate. The module will attempt to decode the barcode assuming it is UPC. If it is not UPC, then decoding
will be attempted assuming the barcode is in Codabar. If the barcode is not Codabar, then the barcode will be considered invalid. Here are the tasks in the module:

1. A task will be created to decode UPC by checking pulse width lengths, which represent bar and space widths, to see if they are valid with a certain amount of error in lengths allowed. High and low pulse width lengths, which represent bar and space widths, will be represented by variables. Each time the barcode wand output is sampled, a variable representing a low pulse or a high pulse will be incremented. When the output switches from low to high or high to low, then another variable will begin to be incremented. Once the data for 2 bars and 2 spaces has been recorded, then it will be decoded into the number it represents; this will be done before sampling for the next number begins. If the first set of data doesn’t represent a number, then this algorithm will stop running. The check-digit calculation, which serves as an error detection test, will be executed once all twelve numbers have been decoded. The check digit is the last digit in the UPC barcode number and it is a function of the rest of the numbers in the barcode number.

2. A task will be created to decode Codabar by checking pulse width lengths for the same reason as in UPC decoding. Similar to UPC decoding, high and low pulse width lengths will be represented by variables and each character will be decoded before sampling for the next character begins. The decoded barcode will be checked to see if A9 are the first 2 characters and A the last character in the barcode to serve as an error detection test.

3. A task will be created to determine if UPC decoding and/or Codabar decoding should be attempted. If both types of decoding should be attempted, then it will initiate the task switch. The user interface description describes when the attempts for both types of decoding should be made.
MAIN- this will be the main module of the program. This module will contain the user interface code and the read/write functions to the serial EEPROM. A task will be created to implement the user interface state machine.

EEPROM_ACCESS- this module will be in charge of changing the nonvolatile information. This module will determine if the user has access to view and/or change information by analyzing the user’s barcode (on the student card) if the user scanned a barcode. Here are the tasks in the module:

1. A task will be created to search for product information given the decoded UPC number.
2. A task will be created to determine user access to information given the decoded Codabar number.

KEYPAD- this module will monitor keypad activity. A task will be created to determine number, letter, or dot selection from the keypad since these characters share buttons.

User Interface Description

The user interface layout, which consists of the LCD, keypad, and barcode wand, are shown in the product sketch in figure 2. The barcode wand will require the user to move the wand across a barcode at a close to constant speed in either direction along a barcode. The user will know when to use the barcode wand based on the system state that is displayed on the LCD.

The barcode scanner system state diagram is shown in figure 4. The display for the RESET state will be this:

| 1 BUY 2 TABS |
| 3 ADMIN  |
When the user presses 1, a scan product prompt will be displayed. When a product is scanned, product price information will be displayed; this is an example of product price information being displayed:

<table>
<thead>
<tr>
<th>NAME: DR. PEPPER</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRICE: $0.50</td>
</tr>
</tbody>
</table>

This is an example of when a user starts a transaction total by pressing the Add button:

<table>
<thead>
<tr>
<th>TRANSACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBTOTAL: $0.50</td>
</tr>
</tbody>
</table>

The user can add to the transaction by scanning another product and pressing the Add button, which adds to the transaction subtotal.

When a user presses 2 while at the RESET state, a scan card prompt will be displayed. When an unauthorized Western ID is scanned, ‘Access denied’ will be displayed for a short period then the RESET state will reappear. When a user scans a valid student ID, the name of the account owner and the tab balance will be displayed; this is an example:

<table>
<thead>
<tr>
<th>JOHN DOE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BALANCE: $5.60</td>
</tr>
</tbody>
</table>

The user can add to the user’s tab in the same manner as the regular transactions are executed, except for the information changes which are shown in the state diagram. When a user wants to pay the user’s tab, this will be the payment prompt displayed:

<table>
<thead>
<tr>
<th>ENTER TAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAYMENT:</td>
</tr>
</tbody>
</table>
The dollar value should be entered followed by the Enter button.

When a user presses 3 while at the RESET state, a scan card prompt will be displayed. When an unauthorized Western ID is scanned, ‘Access denied’ will be displayed for a short period then the RESET state will reappear. When the administrator scans an administrator ID, this option will be displayed:

1 YOUR TAB
2 OTHER INFO

If the administrator presses 1, then the normal tab account function will start; if the administrator presses 2, then this option is displayed:

1 PRODUCTS
2 ACCT 3 REVENUE

If the administrator presses 1, then product options will be shown in this format:

1 TALLIES
2 CREATE/DELETE

If the administrator presses 1, then the product tallies will be shown one at a time on the screen in this format:

NAME: DR. PEPPER
# SOLD: 25

By pressing the arrow buttons, the administrator can switch to other products within the list; by hitting the delete button when any product tally is displayed, all of the product tallies are reset.

If the administrator chooses to create or delete products by pressing 2, then this option will be displayed:
When the administrator decides to add a product to the list by pressing the Add button, there will be a prompt to scan the new product. Then the name with a maximum of 10 characters and the price will be entered by the administrator followed by the Enter button. To delete products, the administrator will press the delete button, which will lead to the product list. The arrow buttons will be used to switch between the products; when the desired product is selected, hitting the delete button will remove that individual product from the system.

When the administrator decides to access account information by pressing 2, this option will be displayed:

```
1 TABPAY 2 TAB
3 ADMIN
```

By pressing 1, the tab accounts will be viewed one at a time and the administrator can switch to other accounts by using the arrow buttons. When a tab account is selected by hitting the Enter button, the administrator will be prompted to enter the tab payment for that account. By pressing 2, this option will be displayed:

```
ADD TABS
DELETE TABS
```

Adding a tab account will be done in the same manner as adding products. The student ID card will be scanned and then the account owner name with a maximum of 16 characters will be entered followed by the Enter button. Deleting tab accounts will be done in the same manner as deleting products; the administrator will select the account by searching through the list of accounts and hitting the delete button.
By pressing 3 at the account information option prompt, this option will be displayed:

```
ADD ADMIN
DELETE ADMIN
```

Adding and deleting administrator accounts will be done in the exact same manner as adding and deleting tab accounts.

When the administrator decides to access revenue information by pressing 3 at the appropriate prompt, the information will be displayed. This is an example of the information:

```
TOTAL $: 67.75
TAB $: 55.60
```

By hitting the delete button when this information is displayed, the total revenue will be reset.

The total tab revenue can only be changed by changing the individual tab balances.

**Development Plan**

Project development this quarter has consisted of defining the project and gathering the necessary hardware for the project. If I complete the software for the project quick enough then a function that allows an administrator to create another administrator account will be added. To prepare for programming the software, I will start studying Part 4 of the Embedded Microcontrollers text to learn how to program in C near the close of fall quarter and into winter break and winter quarter. The main ideas in C that I must concentrate on learning are modules, start-up and initialization, and using the preemptive kernel MicroC/OS-II. There are no potential problems for delivery of parts. Here is the weekly schedule of project development tasks:

**Fall Quarter**

- **Week 9:** Finish the preliminary project description and order some parts.
- **Week 10:** Study the Embedded Microcontrollers text.
Week 11: Work on the final project description.

Week 12: Finish the final project description.

Winter Break

Week 1: Interface the serial EEPROM with the microcontroller and study the Embedded Microcontrollers text.

Week 2: Study the Embedded Microcontrollers text.

Week 3: Start writing the UPC decode task and interface the barcode wand with the microcontroller for testing.

Winter Quarter

Week 1: Continue the UPC decode task.

Week 2: Finish the UPC decode task and start the Codabar decode task.

Week 3: Continue the Codabar decode task.

Week 4: Interface the LCD and keypad with the microcontroller and work on the keypad task for characters to share buttons.

Week 5: Continue the keypad task.

Week 6: Work on the LCD functions.

Week 7: Work on the LCD functions.

Week 8: Work on the product information search task and user access task and the information change/reset task.

Week 9: Work on the administrator information creation/deletion tasks.

Week 10: Work on translating user input to state change commands.

Week 11: Work on translating user input to state change commands.
Spring Break

Week 1: Work on translating user input to state change commands.

Spring Quarter

Week 1: Work on translating user input to state change commands.

Week 2: Configure MicroC/OS-II to implement these state changes and task switching.

Week 3: Configure MicroC/OS-II to implement these state changes and task switching.

Week 4: Configure MicroC/OS-II to implement these state changes and task switching.

Week 5: Configure MicroC/OS-II to implement these state changes and task switching.

Week 6: Configure MicroC/OS-II to implement these state changes and task switching.

Week 7: Configure MicroC/OS-II to implement these state changes and task switching.

Week 8: Left open for expected delays.

Week 9: Left open for expected delays.

Week 10: Put the final touches on the project and demonstrate the project.

Development Software and Hardware

Development of the Barcode Scanner System will occur in ET 340, which is the Engineering Technology (Etec) Lab. The hardware that will be used to develop the project will include this lab equipment in the Etec lab: a power supply, digital multimeter, digital signal oscilloscope, PC, Noral Debugger Pod, and soldering iron. The software programs that will be used to develop the project which is in the Etec lab are these: the Introl C Compiler, CodeWright 6.0, and Flex BDM HC12 debugger software.
Demonstration Prototype and Materials

The demonstration prototype will include the microcontroller development board in the design. The hardware components will be soldered onto the unused space on the development board.

Demonstration of the project will occur at the IEEE snack counter in ET 340. The materials necessary to demonstrate the project are the products at the snack counter, my student ID card, and another student’s ID card. The snack counter product information will be programmed into the project to demonstrate the operation of the project. My student ID card will demonstrate the use of administrator functions and the other student’s ID card will demonstrate tab account creation and regular tab functions.

Electrical Specifications

Project Specifications:

- Barcode size range: 2 – 4 cm
- Scan resolution: 7.95 mils
- Scan speed range: 5 – 50 in/s
- Sweep rate range: 8 – 10 kHz

Power requirements:

- Wall Transformer
- Total worst case power dissipation: 490mW

Special environmental requirements:

- Operating temperature range: 40 - 80°F

PCB size limit: height- 1in., length- 5in., width- 4in.
# Parts List

<table>
<thead>
<tr>
<th>Part</th>
<th>Model no.</th>
<th>Source</th>
<th>Quantity needed</th>
<th>Price</th>
<th>Power Dissipation</th>
<th>Lead Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>2x16 LCD</td>
<td>DMC16207H</td>
<td>Digikey</td>
<td>1</td>
<td>11.76</td>
<td>17.5mW</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Serial EEPROM</td>
<td>25LC640</td>
<td>Digikey</td>
<td>1</td>
<td>0.8</td>
<td>25mW</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Barcode Wand</td>
<td>LA119-15-I00C002</td>
<td>Barcodes, Inc.</td>
<td>1</td>
<td>51.98</td>
<td>100mW</td>
<td>2 weeks</td>
</tr>
<tr>
<td>6 pin DIN connector</td>
<td>2DJ-0196PA02</td>
<td>Digikey</td>
<td>1</td>
<td>0.84</td>
<td></td>
<td>2 weeks</td>
</tr>
<tr>
<td>16 key keypad</td>
<td>96BB2-056-F</td>
<td>Digikey</td>
<td>1</td>
<td>9.653</td>
<td></td>
<td>2 weeks</td>
</tr>
<tr>
<td>Microcontroller</td>
<td>68HC912B32</td>
<td>Digikey</td>
<td>1</td>
<td>14.3382</td>
<td>225mW</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Reset IC</td>
<td>MAX6314</td>
<td>Maxim</td>
<td>1</td>
<td>0.99</td>
<td>66uW</td>
<td>1 week</td>
</tr>
<tr>
<td>16 MHz Crystal</td>
<td>HCM49-16.000MABJT</td>
<td>Digikey</td>
<td>1</td>
<td>1.13</td>
<td></td>
<td>2 weeks</td>
</tr>
<tr>
<td>10k Resistors</td>
<td>CFR-12JB-10K</td>
<td>Digikey</td>
<td>5</td>
<td>0.056</td>
<td>12.5mW</td>
<td>2 weeks</td>
</tr>
<tr>
<td>enclosure</td>
<td>PT-11557-G</td>
<td>Digikey</td>
<td>1</td>
<td>7.2</td>
<td></td>
<td>2 weeks</td>
</tr>
<tr>
<td>power switch</td>
<td>ESB-32101A</td>
<td>Digikey</td>
<td>1</td>
<td>2</td>
<td></td>
<td>2 weeks</td>
</tr>
<tr>
<td>5V regulator</td>
<td>UA78L05CD</td>
<td>Digikey</td>
<td>1</td>
<td>0.32</td>
<td>54mW</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Wall transformer</td>
<td>DPD090015-P7N-DK</td>
<td>Digikey</td>
<td>1</td>
<td>2.6892</td>
<td></td>
<td>2 weeks</td>
</tr>
<tr>
<td>0.33µF capacitor</td>
<td>ECJ-0EB1E331K</td>
<td>Digikey</td>
<td>1</td>
<td>0.205</td>
<td></td>
<td>2 weeks</td>
</tr>
<tr>
<td>0.1µF capacitor</td>
<td>08052R104K8B20D</td>
<td>Digikey</td>
<td>1</td>
<td>0.084</td>
<td></td>
<td>2 weeks</td>
</tr>
</tbody>
</table>