SmartCart: Critical Design Review

Pallavi Jain
Vivian Vasquez
Peter Nguyen
Deniz Kaplan
Product Definition

An interactive touchscreen display affixed onto the handlebars of a shopping cart which allows users to:

- Scan items to a checkout list
- Find aisle locations of items
- Search for item availability
- Obtain item coupons

SmartCart is a smart shopping system designed for consumer convenience and can be applied to:

- Supermarkets
- Department Stores
- Anything that uses shopping carts
Product Description

COUPONS

ADVERTISEMENTS
High Level Block Diagram
LCD Touchscreen

Capacitive Touch Panel
- Prefers 3.0V for logic
- I2C (mapped to VICVectAddr9)
- 6-pin Connector

TFT
- Prefers 3.3V for logic
- Backlight needs 19.2V
- SPI
- 54-pin Connector
**Purpose**: Uses radio frequency communication to identify current aisle location. Interrupt driven (mapped to VICVectAddr29).

**13.56 MHz RFID Mifare Read / Write Module**:
- UART interface
- 5V power supply
- PCB antenna required
PCB Antenna/ RFID Tags

**Purpose**: Allows communication between smart system and tag

**13.56 MHz Mifare PCB Antenna**
- 55mmx55mm
- 70-80mm read range

**Mifare 1K Card**
- NXP S50 Card
- R/W 1K Bytes
Purpose:
Scan each item's barcode and transfer to the RS232 connector, level shifter, and then processor for identification and addition to the "shopping cart" screen of the user interface. Interrupt driven (mapped to VICVectAddr7).

Part:
- ID TECH's "Econoscan ii"
- DB9-DCE Interface
- 5V power supply
- Connector and power supply required
Level Shifter

**Purpose:** Converts serial data from DB9s to Uart and sends it to either the processor or the RFID module. We will include a multiplexing circuit for this because this part only contains 2 level shifter. We need three level shifters total to upload the bootloader code to the processor, send barcode information to the processor, and transmit commands to the RFID circuit in UART.

**Part:** Maxim’s MAX3233E
-3.3V Power Supply

The cell (right) was taken from the homework 1 level shifter (SOIC127P1030_20_D1280N) because the dimensions are not available for the maxim part.
Memory Management: SDRAM

**Purpose:**
- Store layout map
- Interfaces via external memory controller

**Part:**
- Micron's MT48LC8M16A2 – 2 Meg x 16 x 4 banks
- 3.3V
- 54 pin TSOP
- Synchronous on positive clock edge
- At least 100 microsec use delay
Memory Management: SDCard/Connector

Purpose:
For nonvolatile storage of database and other information. Uses SD/MMC interface on microprocessor. Will use FAT filesystem. Write protect and card detect pins on connector float because if card not inserted and/or locked, card detect and write protect float.

Part: Kyocera Memory Card Connector 5638 Series
Testing: There will be DIP switches connected to external reset (identified as bit one in Reset Source ID register 0xE01F C180) and bootloader circuit input pins. There will also be an LED on the reset output pin. Test headers are connected to just about all pins. Unused pins are NC. Two crystals: 32kHz for real time clock (for timer interrupts) [22pF caps] and 20MHz for main oscillator (requiring oscillation mode and driving phase lock loop/PLL ---CLKSRCSEL 0xE01F C10C Register will have value 01) [18pF caps].

Part: NXP LPC2478
- 208 pins
Power Management

Voltages Needed: 19.2V DC, 5V DC, 3.3V DC, 3V DC
Power Planes: 3.3V DC
Voltages Supplied: 19.2V (to save money on converters) and 5V from external power supplies
Linear Regulators: 3.3V LDO with 5V (500mA max) input and 3.0V LDO with 5V (150mA max) input; Biased with 1uF caps; Enable pins tied high

3V Regulator

3.3V Regulator
Software Overview

- We will use a C compiler and Eclipse as our development environment to implement a FAT filesystem with map information stored in onboard memory and database information stored in the SD Card. Upon reset the shopping list and current aisle is cleared.

Display States:
- **Home State:** will allow transition to four other states. The home state should display a map of the store, indicating the last aisle tagged. RFID data should be on the bus if it’s in continuous read mode and raises an interrupt. The interrupt handler should update the home screen with the new aisle information and change the aisle indicator. UART interrupts will be disabled until in the appropriate state. More options should be available as transparent buttons on the home screen, Search, and Shopping List. A timer interrupt for two minutes should be enabled in the home state and when it times out, ads should be displayed in the second sleep state.
- **Sleep State:** This state should also let RFID data be put on the bus to update the current aisle data in memory. Touching the LCD screen in this state should transition back to the home state.
- **Search State:** Touching Search should transition to a search state where a keyboard graphic would be displayed and allow for character input. Any data from the RFID reader prompted by a new tag will be written to memory so the aisle information can be updated in the background. Character input from the keyboard will run a query on the item database and display a list of items that matched the search keyword. A Search button will also be displayed and a back button.
Software Overview

- **Item State:**
  Selecting an item from the search list will display a product description with an image, price, information about whether the item is in the cart, and applicable coupons that can be applied to the shopping list individually. A back button will take the user back to the search interface with the list displayed or a home button would transition to the home state. This state can also be accessed from the Shopping List State screen.

- **Shopping List State:**
  UART will be enabled and a list of all the items in the shopping cart will be displayed. A column of checkboxes, item names, and prices will be displayed. The checkboxes will allow mass removal of items. Touching an item, except for the checkbox, will highlight it and a View button will allow the user to go to the aforementioned product description screen and allow for the same functionality. UART1 will be interrupt-driven since it’s connected to the barcode scanner. Scanning an item in the list screen will raise an interrupt and the handler will add the product to the list.
Software Overview

**Home Screen**
- You are in aisle 6 - Produce and frozen goods
- Search
- Shopping List

**Search Screen**
- Keyword
- List of items pertaining to keyword
- Search
- Return Home

**Item Screen**
- Image
- Item name
- Description
- Price
- Location
- Coupon (if available)
- apply
- Return to Search
- Return Home

**Shopping List**
- Remove?
- Item Name
- Price
- Kellogs Corn Flakes $5.99
- Wonderbread $3.29
- Flamin Hot Funyuns $1.00
- Remove items
- Return Home
- View item
## Software Overview

### Command Code Table for SM130:

<table>
<thead>
<tr>
<th>Code</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x80</td>
<td>Reset</td>
<td>Resets the Module</td>
</tr>
<tr>
<td>0x81</td>
<td>Firmware</td>
<td>Reads the Firmware Revision of the Module</td>
</tr>
<tr>
<td>0x82</td>
<td>Seek for Tag</td>
<td>Continuously checks for presence of a tag</td>
</tr>
<tr>
<td>0x83</td>
<td>Select Tag</td>
<td>Selects a Tag</td>
</tr>
<tr>
<td>0x84</td>
<td>NA</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>0x85</td>
<td>Authenticate</td>
<td>Authenticates the selected Block</td>
</tr>
<tr>
<td>0x86</td>
<td>Read Block</td>
<td>Reads from the specified Block</td>
</tr>
<tr>
<td>0x87</td>
<td>Read Value</td>
<td>Reads from a Value Block</td>
</tr>
<tr>
<td>0x88</td>
<td>NA</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>0x89</td>
<td>Write Block</td>
<td>Writes the data to the specified block</td>
</tr>
<tr>
<td>0x8A</td>
<td>Write Value</td>
<td>Formats and Writes a Value block</td>
</tr>
<tr>
<td>0x8B</td>
<td>Write 4 Byte Block</td>
<td>Writes 4 byte data to Mifare Ultralight block</td>
</tr>
<tr>
<td>0x8C</td>
<td>Write Key</td>
<td>Writes the Key to the EEPROM of the MFRC530</td>
</tr>
<tr>
<td>0x8D</td>
<td>Increment</td>
<td>Increments a value block</td>
</tr>
<tr>
<td>0x8E</td>
<td>Decrement</td>
<td>Decrements a value block</td>
</tr>
<tr>
<td>0x8F</td>
<td>NA</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>0x90</td>
<td>Antenna Power</td>
<td>Switches ON or OFF the RF field</td>
</tr>
<tr>
<td>0x91</td>
<td>Read port</td>
<td>Reads from the Input port</td>
</tr>
<tr>
<td>0x92</td>
<td>Write Port</td>
<td>Writes to the Output port</td>
</tr>
<tr>
<td>0x93</td>
<td>Halt</td>
<td>Halts the PICC</td>
</tr>
<tr>
<td>0x94</td>
<td>Set Baud Rate</td>
<td>Sets the new baud rate</td>
</tr>
<tr>
<td>0x95</td>
<td>NA</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>0x96</td>
<td>Sleep</td>
<td>This command puts SM130 in sleep mode</td>
</tr>
</tbody>
</table>

### Frame:

1. **Header**: 0xFF
2. **Reserved**: 0x00
3. **Length**: Command and the Data bytes
4. **Command Code**
5. **Data**: W/R to 1K Byte tag
6. **CSUM**: Check validity of the packet and to trap any data corruption.
Flow Diagram to perform operations on tag

1. POR
   - Select is required again if tag had left the RF field or halted

2. SELECT
   - Authentication required if sector or access condition is changing

3. AUTHENTICATE
   - No need to authenticate again if sector is not changing provided that the block(s) has same access conditions with previous operations

4. MEMORY OPERATIONS
   - Read
   - Write
   - Increment Value
   - Decrement Value
Software Overview

Exception Generation

<table>
<thead>
<tr>
<th>Address</th>
<th>Exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000 0000</td>
<td>Reset</td>
</tr>
<tr>
<td>0x0000 0004</td>
<td>Undefined Instruction</td>
</tr>
<tr>
<td>0x0000 0008</td>
<td>Software Interrupt</td>
</tr>
<tr>
<td>0x0000 0018</td>
<td>IRQ</td>
</tr>
</tbody>
</table>

Peripheral Register Addresses

- 0xE000 4000  Timer 0
- 0xE000 8000  Timer 1
- 0xE000 C000  UART0
- 0xE001 0000  UART1
- 0xE007 8000  UART2
- 0xE001 C000  I2C0
- 0xE002 0000  SPI
- 0xE002 4000  RTC
- 0xE008 C000  SD/MMC Card Interface
- 0xE01F C000  System Control Block

Clocking Registers

- PLL Control register (for PLL Control)
  PLLCON - 0xE01F C080
- PLL Configuration register (for configuring all multipliers and dividers)
  PLLCFG - 0xE01F C084
  \[ FCCO = \left(\frac{2 \times M \times FIN}{N}\right) \]
  N and M configurable in register
- CPU Clock Configuration register
  CCLKCFG - 0xE01F C104
- Peripheral Clock Selection registers 0 and 1
  PCLKSEL0 - 0xE01F C1A8
    - 3:2 PCLK_TIMER0
    - 5:4 PCLK_TIMER1
    - 7:6 PCLK_UART0
    - 9:8 PCLK_UART1
    - 15:14 PCLK_I2C0
    - 17:16 PCLK_SPI
    - 19:18 PCLK_RTC
  PCLKSEL1 - 0xE01F C1AC
    - 17:16 PCLK_UART2
    - 19:18 PCLK_UART3
    - 29:28 PCLK_SYSCON

Will use default round robin scheduling or priority of CPU, GPDMA, AHB, LCD
Various approaches involving RS232, LEDs, and Berg test headers
- All pins interfacing to MCU connected to test headers
- RS232 connection to a terminal
- LED outputs on resets, invalid pins, and output OK and ERR pins
  Biased with 65Ohm (3.3V supply) or 150Ohm (5V supply) with 20mA driving
- DIP switches on some pins (e.g. reset)

Switch Debouncing
\[ R = 10k\Omega \]
\[ C = 0.1\mu F \]
Schematic (Bypass Caps)