Senior Project Description
12/9/03

Author: Derek Valleroy
Project: Scrolling Message Display
Draft: Final Draft
Course: ETEC 471
Instructor: Todd Morton
Institution: Western Washington University
**Introduction:**

Information can be exchanged in a variety of ways. In the past two decades we have seen an explosion in technology dedicated to visual communication. Information is now more accessible and appealing through televisions, computers, and video displays. I propose to build a scrolling message display for the IEEE club. Messages regarding the IEEE club are usually communicated through flyers and informal notes on the laboratory white board. Through the scrolling message display, this communication process can be improved. Club meeting times and important deadlines will be delivered more effectively through an eye-catching medium.

**Hardware Description:**

The scrolling message display will be a sign with animated text that is intended for indoor use. A maximum of ten characters can be displayed on this sign at one time and by scrolling through the characters, the user can display messages that are much longer. This display will be viewable up to a hundred feet away, which roughly corresponds to a character height of two inches. The scrolling message display will be a “stand alone” project, so after a message is saved in memory the sign can be powered down and placed elsewhere. This sign can be mounted on a wall or placed anywhere that has access to a wall outlet. The maximum dimensions of the sign within the case will be 4’’ X 20’’ X 4’’. The final product should resemble the sign depicted in Figure 1.
Detailed Functional Description:

The scrolling message display will allow the user to type in any alphanumeric message and see it displayed through an array of red LED’s. A system block diagram illustrating this process is depicted in Figure 2.

The LED’s will be arranged in an array of seven rows by sixty columns. An LED driver such as the MAX6952 will be connected to the SPI of a microcontroller.

This project will be an embedded system that uses Motorola’s MC9S12DP256B microcontroller (Star12). The Star12 was selected for its large amount of byte erasable EEPROM and for its versatility. Figure 3 is a detailed functional block diagram that depicts the resources that will be needed on the microcontroller.
The SCI bus is used to communicate data between the user and a PC terminal program via the RS232 interface. Data will be sent through port PS1 (TXD0) and received through port PS0 (RXD0). The SPI bus communicates the display data to the LED driver through the four-wire serial interface. The three outputs that are sent across this interface are: *Data OUT* (MISO0), *Clock* (SCK0), and *Chip Select* (/SS0). The BDM will only be used to upload the final program into the microprocessor’s Flash memory. A 16MHZ crystal will be used to clock the Star12. Reset circuitry will be used to ensure that the microcontroller operates at a steady supply voltage of 5 volts DC.
The memory scheme of the Star12 includes 256K Bytes of Flash EEPROM, 4K Bytes of byte-erasable EEPROM and 12K Bytes of RAM. Messages will be stored as ASCII characters in the byte-erasable EEPROM. The user will have the option to create and save five messages, each with a total length of two-hundred characters. If all five messages are used, a total of 1K Byte of EEPROM is required. The main program will be stored in the Flash EEPROM. Every character within a message will be converted into a 6-byte bit-map and then stored in RAM. For the maximum message length, a total of 1.2K bytes of RAM will be needed for each message.

The scrolling message display is intended for indoor use, so a wall transformer will be used to power the microcontroller and all external hardware. This wall transformer will have an input of 120V-AC, and an output of 5V-DC. A maximum output current of 1.0A will be enough to power all system hardware. A more detailed description of the maximum power dissipation of this system can be found in the Electrical Specifications portion of this report.

Software Description:

The scrolling message display will require software to manipulate the user defined messages into signals appropriate for the LED drivers. This software will be written in C programming language and a number of modules will be used to perform the various tasks involved in the software. The MicroC/OS-2 preemptive kernel will be used to switch between tasks. A small amount of the code will be written in assembly language where it is appropriate. The modules that will be used in the software are KERNEL, MAIN, STORE_STRING, BITMAP, SPI_DISPLAY and BASICIO.

- The KERNEL module will incorporate MicroC/OS-2 for task management.
• The MAIN module will control what is being displayed on the terminal program. This module also allows the user to decide how many messages will be used, how fast the display will scroll, and the desired intensity of the sign.

• The STORE_STRING module will store the five user defined messages in the byte-erasable EEPROM. These messages will be stored as ASCII characters which will later be decoded by the BITMAP module. STORE_STRING will also maintain the order in which the ASCII characters were entered by the user.

• The BITMAP module is used to convert the string of ASCII characters into a bitmap. Each character will be created as 7 bits tall by 5 bits wide. An additional bit will be added to the width of each character to create spacing between individual characters. These characters will be stored in the microcontroller’s RAM. This module will be written by myself, however it will be highly advantageous if I can find a similar module that has already been written.

• The SPI_DISPLAY module will be responsible for transmitting messages to the LED drivers. This module will continuously be sending words to the drivers that contain a command byte and a data byte. The data bytes will be sent from the bitmap that is stored in RAM.

• The BASICIO module will be used exclusively in the user interface. BASICIO will take care of user input from the keyboard and how it is displayed in the terminal program. This module was written by Todd Morton.

User Interface:

The user interface for this project will consist of a terminal program on a PC that uses a keyboard for user entry. The terminal program will use the VT100 terminal protocol. A 9-pin
serial plug (DB9) will be used to connect the PC with the RS232 interface on the microcontroller. An example of the interface display that the user will see is depicted in Figure 4.

![Figure 4](image_url)

**Figure 4.(Example of the terminal program user interface)**

Upon power-up, all selected messages will automatically start scrolling across the LED display. This allows the user to place the sign in a remote location, regardless of whether it is connected to a PC or not. There will be an initial reset sequence that the sign will go through before it starts scrolling through the user defined messages.
When the sign is connected to a PC, the user will be given the eight options that are depicted in Figure 4. The user can select an option by entering the appropriate number, followed by a carriage return. The first five options allow the user to edit the five available messages. When any of these five options are selected, the user will be prompted to enter any message created from ASCII characters. If the user exceeds the 200 maximum characters, the last character will keep writing over itself. When a carriage return is detected, the user will then be prompted to enter a title for the new message (see Figure 5 for an example sequence).

```
Please enter an option
> 3
Type in your message (maximum of 200 characters)
> Example message. ABC123#$%?
Enter a title for you message (10 character max)
> ExampleMes
```

Figure 5.(Example sequence for message editing)

After a carriage return is detected for the title prompt, the message will be automatically saved and the main option screen is displayed again with the updated title for the new message (see Figure 6).
When the user selects option 6 (scroll speed), there will be a message that prompts the user to select a speed by entering a number between ‘0’ and ‘5’. A ‘5’ represents the fastest scroll speed and a ‘0’ represents no movement at all. After a number is entered and a carriage return is detected, the speed will be saved and the main option screen gets displayed again.

When the user selects option 7 (sign intensity), there will be a message that prompts the user to select a number between ‘1’ and ‘15’. A ‘1’ represents the dimmest intensity and a ‘15’ represents the brightest intensity. After a number is entered and a carriage return is detected, the intensity will be saved and the main option screen gets displayed again. When the user selects option 8 (message order), there will be a message that prompts to enter in the order of the messages. The order is selected by entering in the respective numbers of the saved messages in the order that is desired (see Figure 7 for example sequence).
After the sequence of numbers is entered and a carriage return is detected, the order is saved and the main option screen is displayed again. In the example depicted in Figure 7, the third message will be displayed first, then the second message, then the first. If there is no entry for a particular message, that message will not be displayed on the sign. If the user types in any character that is not valid for a particular option, a default message will be displayed and the user will be prompted to enter another selection.

**Development Plan:**

At this point in the development phase, all of the hardware has been selected and an overview of the software has been conceptualized. Two samples of the LED driver IC’s have been ordered but I still need to order one more to complete this project. The only other hardware devices that need to be ordered are the LED matrices. These matrices should pose no delivery problems unless all vendors run out of their stock. If I can’t find a vendor with LED matrices in stock, then I will need to have matrices custom made and delivered upon completion.
The final hardware design is almost completed and the only remaining tasks are to lay out the circuit on a PCB (Printed Circuit Board) and build the case. A majority of my time will be spent learning the C programming language and writing the software. In winter quarter of 2004, I will be taking an Embedded Systems class, which will provide me with the knowledge necessary to develop the software. A weekly schedule for the tasks to be completed during winter and spring quarters is depicted in Figure 8.

WINTER QUARTER (2004)

Week 1 – Order remaining parts
Week 2 – Design PCB layout
Week 3 – Design PCB layout
Week 4 – Construct circuit and mount it on the PCB
Week 5 – Test circuit layout and verify that all parts have been wired properly
Week 6 – Build the case and mount the PCB inside of it
Week 7 – Research pre-written modules that can be used
Week 8 – Begin software design
Week 9 – Modify MicroC/OS-2 module
Week 10 – Modify MicroC/OS-2 module
Week 11 – Complete all hardware.

SPRING QUARTER (2004)

Week 1 – Write and debug the MAIN module
Week 2 – Modify the BASICIO module
Week 3 – Write and debug the STORE_STRING module / Hardware Review
Week 4 – Design the bit-map used in the BITMAP module
Week 5 – Write or modify the BITMAP module
Week 6 – Write the SPI module
Week 7 – Integrate all software modules
Week 8 – Upload final program to the microcontroller
Week 9 – Work out any bugs that may occur
Week 10 – Code review
Week 11 – Project demonstration

Figure 8. (Weekly schedule for project development)

Development Hardware and Software:

This project will be primarily developed in the Engineering Technology lab (ET 340) at Western Washington University. This lab contains all of the hardware and software development tools that will be needed. The types of hardware that will be used include bench power supplies, mixed signal oscilloscopes, PC’s, Noral Debugger Pods, soldering irons and PCB production equipment. The development software that will be used in this lab include the Introl C Compiler, CodeWright, schematic capture programs (Eagle), and BDM HC12 debugger software. The only hardware that will not be developed in this lab is the case of the sign. The case will be constructed at my private residence.

Demonstration Prototype and Materials:

The demonstration prototype will be enclosed in a wooden case that reflects the maximum dimensions of the final product. Within this case will be the main PCB and the Star12 evaluation board. The main PCB will have the driver chips and LED matrices soldered directly onto it. The Star12 evaluation board will be mounted to the main PCB and wires will be used to
connect it with the electrical components on the main PCB. In a stand-alone final product, the Star12 would be combined with the other hardware on a single board.

To demonstrate this prototype, a PC will be needed to edit and save any of the five messages. After a message is saved, the sign can be powered down and placed elsewhere. When the sign is powered up again, the saved messages will automatically start scrolling across the sign.

**Electrical Specifications:**

**LED Display**

- Number of saved messages: 5
- Individual message length: 200 characters
- Maximum number of characters displayed at one time: 10
- Character height: 2”
- Character length: 1.5”
- Valid characters: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz1234567890,.?:;’()!@#$%^&*-_+=
- Scroll speeds: Static, 1in/sec 2in/sec, 3in/sec, 4in/sec, 5in/sec
- Minimum segment intensity: 400uCD

**Serial Specifications**

- Interface: RS232
- Protocol: Asynchronous Start/Stop
- Baud rate: 9600
- Data: 8-bit
- Parity: None
Power Source

Type: Wall transformer

Input: 120V AC, 60Hz

Output: 5V DC, 1.0A

Worst case power dissipation: 1.634Watts

Special Environment Considerations

Application environment: Indoor use only

Operating temperature range: 0º to +50º C

PCB Size Limits

Length: 18”

Width: 2”

Height: 2”
### Preliminary Parts List:

<table>
<thead>
<tr>
<th>Description</th>
<th>Part #</th>
<th>Qty</th>
<th>Source</th>
<th>Price</th>
<th>Power Dissipation</th>
<th>Lead Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller</td>
<td>MC9S12DP256B</td>
<td>1</td>
<td>Motorola</td>
<td>$20.65</td>
<td>65mA</td>
<td>1 week</td>
</tr>
<tr>
<td>LED Driver</td>
<td>MAX6952</td>
<td>3</td>
<td>Maxim</td>
<td>$25.95</td>
<td>229.6mA</td>
<td>1 week</td>
</tr>
<tr>
<td>LED Matrix</td>
<td>GMC2975C</td>
<td>12</td>
<td>Mouser</td>
<td>$51.60</td>
<td>31.5mA</td>
<td>4 weeks</td>
</tr>
<tr>
<td>Reset Circuit</td>
<td>MAX6314</td>
<td>1</td>
<td>Maxim</td>
<td>$0.99</td>
<td>12uA</td>
<td>1 week</td>
</tr>
<tr>
<td>Case</td>
<td>NA</td>
<td>1</td>
<td>Personal</td>
<td>$20.00</td>
<td>N/A</td>
<td>1 week</td>
</tr>
<tr>
<td>Crystal Oscillator</td>
<td>CSA309</td>
<td>1</td>
<td>Digikey</td>
<td>$0.65</td>
<td>N/A</td>
<td>1 week</td>
</tr>
<tr>
<td>232 Interface</td>
<td>MAX232A</td>
<td>1</td>
<td>Maxim</td>
<td>$1.85</td>
<td>25uA</td>
<td>1 week</td>
</tr>
<tr>
<td>9-pin serial plug</td>
<td>DB9</td>
<td>1</td>
<td>All electronics</td>
<td>$0.45</td>
<td>N/A</td>
<td>1 week</td>
</tr>
<tr>
<td>Wall Transformer</td>
<td>DTR050150</td>
<td>1</td>
<td>Digikey</td>
<td>$2.50</td>
<td>N/A</td>
<td>2 weeks</td>
</tr>
<tr>
<td>1uF capacitor</td>
<td>N/A</td>
<td>8</td>
<td>Digikey</td>
<td>$0.80</td>
<td>N/A</td>
<td>2 weeks</td>
</tr>
<tr>
<td>47uF capacitor</td>
<td>N/A</td>
<td>3</td>
<td>Digikey</td>
<td>$0.30</td>
<td>N/A</td>
<td>2 weeks</td>
</tr>
<tr>
<td>26pF capacitor</td>
<td>N/A</td>
<td>3</td>
<td>Digikey</td>
<td>$0.30</td>
<td>N/A</td>
<td>2 weeks</td>
</tr>
<tr>
<td>53.6K resistor</td>
<td>N/A</td>
<td>3</td>
<td>Digikey</td>
<td>$0.30</td>
<td>220uA</td>
<td>2 weeks</td>
</tr>
<tr>
<td>4.7K resistor</td>
<td>N/A</td>
<td>3</td>
<td>Digikey</td>
<td>$0.30</td>
<td>540uA</td>
<td>2 weeks</td>
</tr>
<tr>
<td><strong>TOTALS:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Maximum Power Dissipation:</strong></td>
<td>326.9mA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total Cost of Parts:</strong></td>
<td><strong>$126.64</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>